



[Home](#) / [Refactoring](#) / [Code Smells](#) / [Object-Orientation Abusers](#)

Refused Bequest

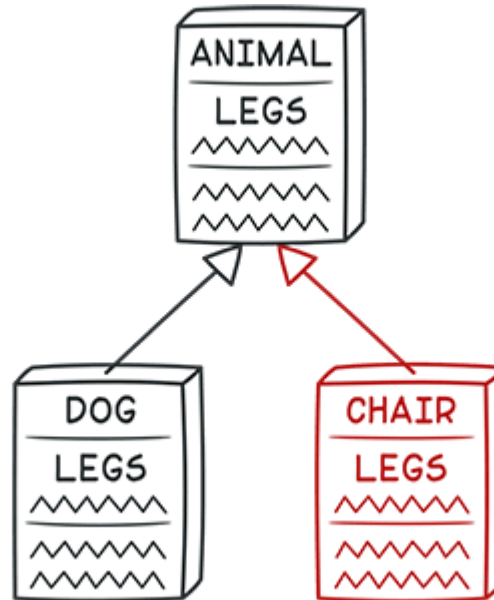
Signs and Symptoms

If a subclass uses only some of the methods and properties inherited from its parents, the hierarchy is off-kilter. The unneeded methods may simply go unused or be redefined and give off exceptions.



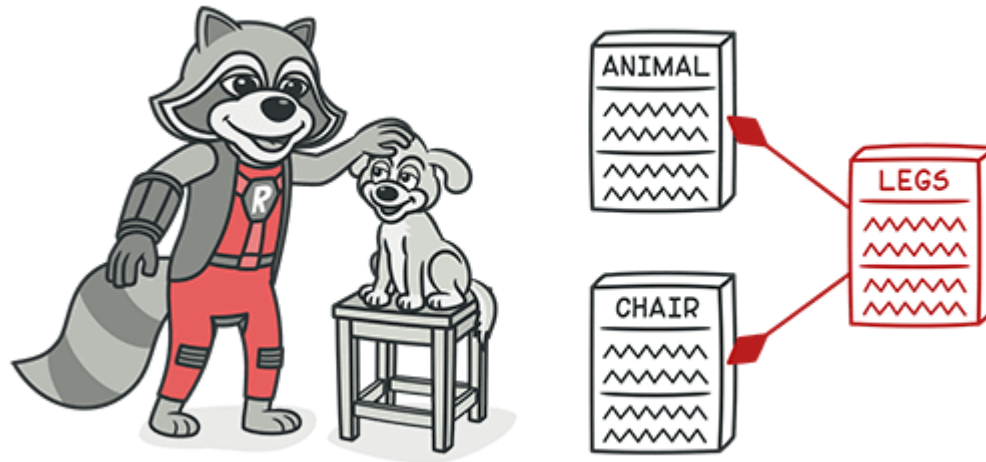
Reasons for the Problem

Someone was motivated to create inheritance between classes only by the desire to reuse the code in a superclass. But the superclass and subclass are completely different.



Treatment

- If inheritance makes no sense and the subclass really does have nothing in common with the superclass, eliminate inheritance in favor of **Replace Inheritance with Delegation**.
- If inheritance is appropriate, get rid of unneeded fields and methods in the subclass. Extract all fields and methods needed by the subclass from the parent class, put them in a new subclass, and set both classes to inherit from it (**Extract Superclass**).



Payoff

Improves code clarity and organization. You will no longer have to wonder why the `Dog` class is inherited from the `Chair` class (even though they both have 4 legs).

