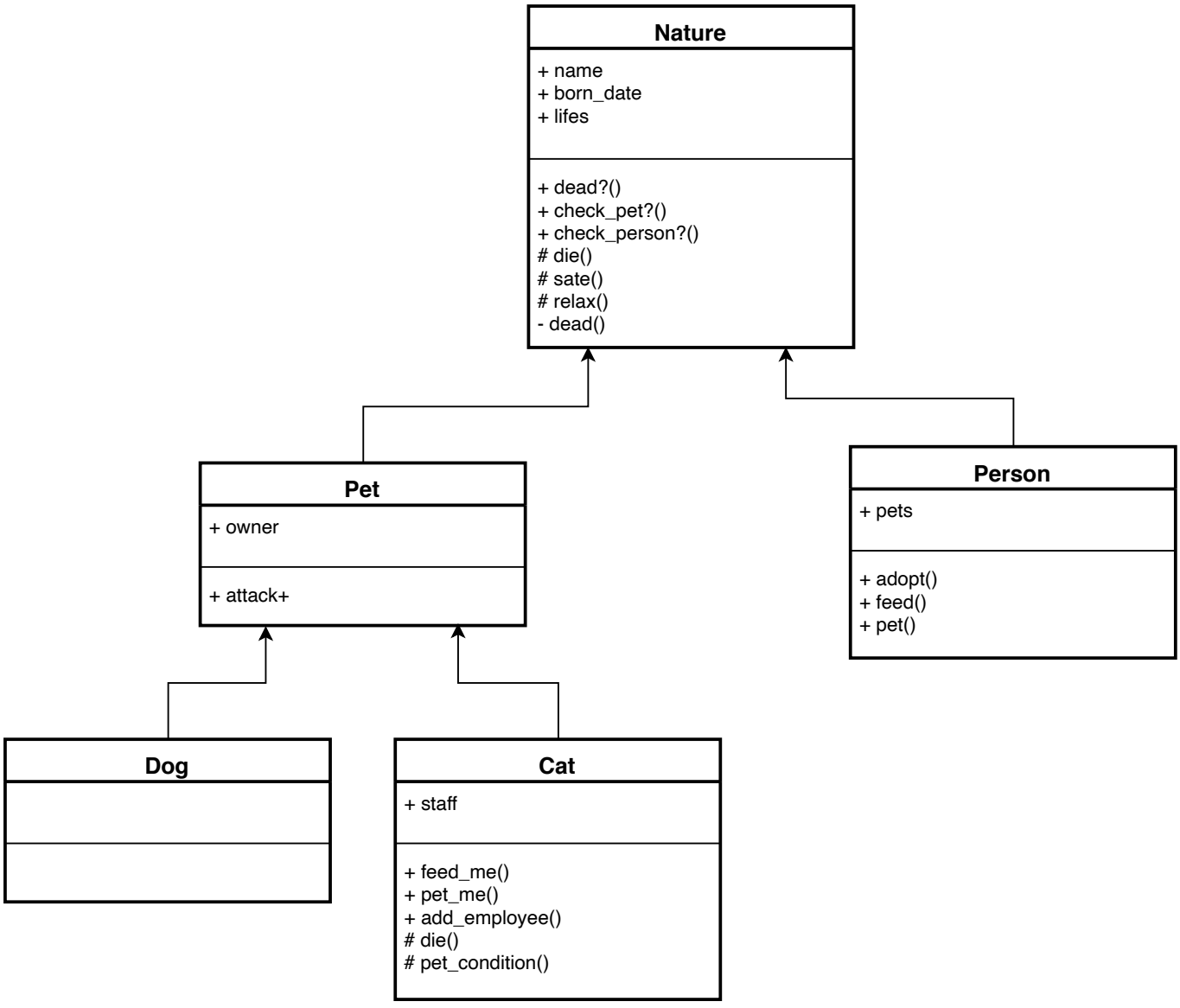
Professor:: Dr. Bernd Kahlbrandt
Author:: Nick Marvin Rattay
Author:: Hani Alshikh

24.11.2019

Vererbung, Assoziationen und Methodensichtbarkeit

für diese Aufgabe wurde die folgende Struktur entworfen



Erläuterung

da es keine Angaben über die Gleichheit gab, wurde entschieden, dass zwei object der Nature Klasse (Tiere, Personen usw...) nur dann gleich sind, wenn sie Objektidentisch sind.

Klasse "Nature"

da eine natürliche Reaktion auf aktivitäten wie essen, stricheln und angreifen (sterben) sinnvoll sei wurde die Klasse "Nature" als Oberklasse vom Leben jeglicher Art extra entworfen.

Alle Reaktionmethoden wie die, sate, relax usw... haben die Sichtbarkeit "Protected", denn sie Folgemethoden und dürfen nicht direkt aufgerufen werden.

die Methode #pet_condition wurde als Hookmethode benutzt, um Fälle zu prüfen, wie Katzen lassen sich nur von ihrem Personal streicheln bzw füttern.

Verwendung vom Set

die Datenstruktur Set wurde für die Sammlung Haustiere bzw Personen verwendet, um sicherzustellen, dass es keine duplizierung geben könnte.

Fragen

- gibt es eine bessere lösung als die Nutzung von #pet_condition?
  unsere erste Idee war die Methoden #sate, #relax protected zu machen, aber dann dürfen Objekte der Klasse Person die Methoden nicht mehr benutzen.
  Also allgemein gefragt, kann man der Aufruf einer Methode nur für bestimmten Klassen erlauben, ohne die Klassen von der gleichen Parentklasse vererben zu müssen?
- kann man ein Objekt "umbringen"? d.h. kann man z.b. ein Exception raisen, wenn ein Objekt gestorben ist, ohne jedes Mal zu prüfen, ob das Objekt gestorben ist?
  unsere Idee war, alle Methoden auf eine Methode (#dead) zu verweisen, in dem wir #alias_method benutzen, wenn das Objekt gestorben ist.

Quellen

- Set
- Test::Unit