Inheritance Practice

Here is an interface that represents a virtual Pet.

```
public class Pet
{
    public void reward(int numTimes)
    {
    }

    public void punish(int numTimes)
    {
    }

    public void act()
    {
    }
}
```

Your task is to create two subclasses for Pet, named Dog and Cat. Each one of them should override the three methods above appropriately, as well as contain a relevant constructor. Here are some more details:

- Each Pet object should keep track of an "amount of happiness" (an integer that starts off as 0) as an instance variable.
- For a Dog, the method `reward` should increase happiness by 3 each time the animal is rewarded. For a Cat, it should only increase happiness by 1 each time the animal is rewarded. (Cats are harder to impress.)
- For a Dog, the method `punish` should decrease happiness by 2 each time the animal is punished. For a Cat, it should decrease happiness by 3 each time the animal is punished. (Cats are more likely to sulk.)
- Create a method `act()` that will do two things: print out to the screen the current happiness value for the animal, and then act accordingly by printing out some kind of animal noise. Regarding the animal noise, be creative: how might a dog or cat act differently depending on the happiness value? What might be different in the rules that you use, between a dog and a cat? There's no "right answer" for how to act (apart from printing out the current happiness value first), but do something interesting that's different between the two.

If you've written your class correctly, the following test code should work if you place it inside an appropriate main method:

```
    Pet fido = new Dog();
    Pet socks = new Cat();

    fido.reward(5);
    socks.punish(3);

    fido.act();
    socks.act();
```