At Generic Pet Store LLC., There are dogs, cats, and snakes. We want to make a small application that reflects this pet store, in order to practice using Interfaces. This is also a good way to wrap your head around the concepts of Inheritance and Polymorphism.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

In a package called com.revature.models (or whatever you want to call your models package), create an Animal Interface. This interface will lay out a few abstract methods that any Class implementing the interface must override. Create the following abstract methods:

-void makeNoise();

-void walk();

-SOME ABSTRACT METHOD OF YOUR CHOICE.

Now, also in the models package, create three concrete (normal) Classes - Dog, Cat, and Snake. Make these classes IMPLEMENT Animal, thus requiring the Class to implement the three abstract methods found in the Animal interface. Check HelloAbstract if you don’t remember how to make a class implement an Interface

In each concrete Class, override the abstract methods of the Animal class however you see fit. To keep it simple, you could just have the methods print something out. For instance:

@Override

void makeNoise() {

//PRINT STATEMENT WOULD GO HERE

}

The easiest way to auto generate the abstract methods is to make a class implement the interface, and mouse over the error that occurs. It’ll give you the option to auto generate the unimplemented methods. Check HelloAbstract for more examples of overridden methods!

Finally, create a Class called Launcher with a main method in the com.revature package

Instantiate a Dog, Cat, and Snake object, and call each method contained in each concrete class.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

This exercise is a good example of Inheritance (in the form of implementing interfaces)

As well as Polymorphism (by creating different forms of Animal, and overriding methods)

Don’t forget to read up on the pillars of OOP! The acronym APIE can be helpful.