CS 116 - Lab 7
All Java files due in Blackboard Assignment by Thursday 3/26 before Lab (10am)

Objectives:

1. (10 points) Design, code and test classes demonstrating inheritance (Greenfoot).
2. (10 points) Design, code and test an abstract class and derived classes demonstrating inheritance and polymorphism.

Tasks:

1. (10 points) Design, code and test classes demonstrating inheritance (Greenfoot).

A. Add another "Animal" subclass of your choosing that will move more quickly/randomly, eat the lettuce (to steal from the turtle), but will not be effected by the snakes. Also make it harder to kill by the turtle by requiring multiple "hits" by the turtle (maybe this new class needs an attribute to count how many times it is hit by the turtle, and an increment method for that attribute). The Turtle can call this method by using (Bug)getOneObjectAtOffse(0,0,Bug.class); to get a reference to this new object when it hits it.

Instances of this new Animal subclass will check its own hit count in act(), and remove itself from the world when it reaches some limit you define (note: One intersection of two objects will actually register as multiple hits, maybe 3 or 4, due to the speed of the act() methods being called).

To define the new subclass and choose the default image right click on Animal class and choose New Subclass, name the class and choose an image. The class shell will be created with an empty Act method.

Remember to update your TurtleWorld prepare() method to add an instance of this new Animal subclass to a random position in the World at the start of the game.

Also update the Turtle act() method to have it attempt to eat instances of this new Animal subclass.

B. We want to keep score in our Trick the Turtle game. Add the Counter class inheriting from Actor using the image and my Counter class.

http://www.cs.iit.edu/~cs116mb/labs/Lab7/counter.png

http://www.cs.iit.edu/~cs116mb/labs/Lab7/Counter.java

The Turtle needs to collaborate with the Counter. We could do this through the World methods, but another way to do this is send the reference of the Counter object to the Turtle object when you construct the Turtle. To do this you will need to do the following:

- add an attribute to the Turtle to store a reference to a Counter object
- add a new constructor for the Turtle that has a Counter object reference as an attribute
- add code to instantiate the Counter in TurtleWorld (before you instantiate the Turtle), place it in a corner, and then call the new Turtle constuctor

Now your Turtle can call the Counter method add(int score) when the Turtle eats a lettuce.

2. (10 points) Design, code and test an abstract class and derived classes demonstrating inheritance

and polymorphism.

A Sphere or Cylinder are both CircleSolid objects, they both have a radius. A Sphere "is a" CircleSolid object and a Cylinder "is a" CircleSolid object. But such a thing as a CircleSolid object does not exist, it is an abstraction we come up with to group 3-D objects. We saw in lecture that Object-Oriented programming supports abstract objects and inheritance. Abstract objects cannot be instantiated (never can be "new"d).

Create an abstract class CircleSolid with the following:

- attribute radius
- a default constructor
- a non-default constructor with a radius as parameter
- setRadius & getRadius & getCircumference & getArea & toString
- an abstract method getVolume

The create a Sphere class that extends CircleSolid class with the following:

- a default constructor
- a non-default constructor with a radius as parameter
- getVolume & toString

The create a Cylinder class that extends CircleSolid class with the following:

- attribute height
- a default constructor
- a non-default constructor with a radius and height as parameters
- setHeight & getHeight
- getVolume & toString

Test with my client program, target output below:

http://www.cs.iit.edu/~cs116mb/labs/Lab7/CircleSolidClient.java

```
CircleSolid: Radius=28.0
Cylinder: Height=35.0 Volume=86205.30241450392

CircleSolid: Radius=6.0
Sphere: Volume=678.5840131753953

Approximate number of spheres in this cylinder=127.03703703703704

CircleSolid: Radius=28.0
Cylinder: Height=35.0 Volume=86205.30241450392

CircleSolid: Radius=6.0
Sphere: Volume=678.5840131753953

class Sphere Radius=6.0 Volume=678.5840131753953
class Cylinder Radius=10.0 Volume=4712.38898038469
class Sphere Radius=20.0 Volume=25132.741228718343
```