Set 7

The source code for the Critter class is in the critters directory
1. What methods are implemented in Critter?
act() getActors() processActors() getMoveLocations() selectMoveLocations() makeMove()
2. What are the five basic actions common to all critters when they act?
getActors() processActors() getMoveLocations() selectMoveLocations() makeMove()
3. Should subclasses of Critter override the getActors method? Explain.
Maybe. For example, crab only gets the actors AHEAD, HALF_LEFT, HALFR_RIGHT. So crab override getActors() method. But chameleon need not override this method because it also gets the actors
around it.
4. Describe the way that a critter could process actors.
Eat all of them.
5. What three methods must be invoked to make a critter move? Explain each of these methods.
getMoveLocations() selectMoveLocations() makeMove()
6. Why is there no Critter constructor?
Because Critter has no special private variable. We needn't create constructor for it. Just derived from the Actor class

Set 8

The source code for the ChameleonCritter class is in the critters directory
1. Why does act cause a ChameleonCritter to act differently from a Critter even though ChameleonCritter does not override act?
Because chameleon overrides two other methods: processActors() makeMove()
2. Why does the makeMove method of ChameleonCritter call super.makeMove?
Chameleon just need to change its direction. And its other behaviors are the same as its super class.
3. How would you make the ChameleonCritter drop flowers in its old location when it moves?

```
  public void makeMove(Location loc) {
    Location oldloc = this.getLocation();
    Grid gr = this.getGrid();
    Flower flower = new Flower(this.getColor());
    setDirection(getLocation().getDirectionToward(loc));
    super.makeMove(loc);
    flower.putSelfInGrid(gr, oldloc);
  }
```

4. Why doesn't ChameleonCritter override the getActors method?
Because chameleon also gets the actors around it, same as critter.
5. Which class contains the getLocation method?
Actor Class
6. How can a Critter access its own grid?
Grid gr = this.getGrid();
gr.get(this.getLocation());

Set 9

The source code for the CrabCritter class is reproduced at the end of this part of GridWorld.
1. Why doesn't CrabCritter override the processActors method?
Because crab also eats the actors as the same as critter class.
2. Describe the process a CrabCritter uses to find and eat other actors. Does it always eat all neighboring actors? Explain.
crab.getActors(); crab.processActors();

Crab get the actors AHEAD, HALF_LEFT, HALF_RIGHT, and then eats them, not all neighboring actors.

3. Why is the getLocationsInDirections method used in CrabCritter?

Because the crab needs to get locations in some specific directions, not all directions.

4. If a CrabCritter has location (3, 4) and faces south, what are the possible locations for actors that are returned by a call to the getActors method?

Location(4, 4) Location(4, 3) Location(4, 5)

5. What are the similarities and differences between the movements of a CrabCritter and a Critter?

Similarities: Both select a random location to move.

Differences: Crab can only move right or left. Critter can move to any neighboring locations.

6. How does a CrabCritter determine when it turns instead of moving?

turn left or right at random.

7. Why don't the CrabCritter objects eat each other?

CrabCritter is also a critter.

```
if(!(a instanceof Rock) && !(a instanceof Critter)) {
  a.removeSelfFromGrid();
}
```