

Part 4:Interacting Objects

Set 7

- 1.act(),getArctors(),processActors(),getMoveLocations,selectMoveLocation(),makeMove();
- 2.getArctors(),processActors(),getMoveLocations,selectMoveLocation(),makeMove();
- 3.Yes.If the new subclass selects its actors from different locations,it should override the getActors method.
- 4.It could eat all of them, it could make them change their color, or it could ask them to move.
- 5.getMoveLocation—return a list of all empty adjacent locaitons around the critter.
selectMoveLocation—select a location from the list returned by getMoveLocation
makeMove—make the critter move to the new location.
- 6.Critter extends Actor and the Actor class has a default constructor.And the default Critter constructor will call super(), which call the Actor default constructor.

Set 8

- 1.Because the chameleonCritter overrides the processActors and makeMove methods.When call act(), the act() will use chameleonCritter's processActors and makeMove to produce different behavior.
- 2.Because it's behavior is same with the Critter, so use super.makeMove is convenient.
- 3.Modify the makeMove method:

```
public void makeMove(Location loc) {  
    setDirection(getLocation().getDirectionToward(loc));  
    super.makeMove(loc);  
    Location oldLoc = getLocation();  
    if (!oldLoc.equal(loc)) {  
        Flower flower = new Flower();  
        flower.pubSelfInGrid(getGrid(), oldLoc);  
    }  
}
```
- 4.Because it processes the same list of actors that Critter does, no new behavior.
- 5.The Actor class.
- 6.By calling getGrid().

Set 9

- 1.Because its processActor behavior is same with the Critter.
- 2.The getActors method only look for neighbors that are immediately in front, right front and left front of the crab critter.Any neighbors found in these locations will be “eaten”.
- 3.The parameter for this method bring in an array of directions.And the getActors() use this method to get the possible neighbors in specific directions.
- 4.(4,3),(4,4),(4,5)
- 5.Similarities:they both randomly choose next location to move, and they don't turn in the direction that they are moving.
Differences:CrabCritter only can move to its left and right and Critter can move to any direction of its adjacent neighbor locations.
- 6.If the parameter loc in makeMove is equal to the current location. It turns instead of moving.
- 7.Because the CrabCritter inherits the processActors from the Critter class.And the method only remove actors that are not rocks and critters.But a CrabCritter is a critter.