

Set 7

1. What methods are implemented in Critter?

The act, getActors, processActors, getMoveLocations, selectMoveLocation and moveMove method are implemented in Critter.

2. What are the five basic actions common to all critters when they act?

The getActors, processActors, getMoveLocations, selectMoveLocation and moveMove method

3. Should subclasses of Critter override the getActors method? Explain.

Yes, when a new critter subclass can select its actors in different style, it could override this method.

4. Describe the way that a critter could process actors.

It could eat, change colors or push them to move.

5. What three methods must be invoked to make a critter move? Explain each of these methods.

The three methods are getMoveLocations, selectMoveLocation and makeMove.

getMoveLocations return all the possible location, which is empty adjacent locations around the critter, to move, and select MoveLocation choose a location from these location list to move, at last, moveMove determine the move style to the choose location.

6. Why is there no Critter constructor?

Critter extends Actor, which has a default constructor. When you don't create a constructor for critter class, Java will write a default constructor for you.

Set 8

1. Why does act cause a ChameleonCritter to act differently from a Critter even though ChameleonCritter does not override act?

Because ChameleonCritter override the processActors and moveMove methods, which is called in act method.

2. Why does the makeMove method of ChameleonCritter call super.makeMove?

The makeMove method of the ChameleonCritter share the same behaviors with Critter's moveMove method except changing the directions of critter to face its new location, so the ChameleonCritter call super.makeMove to diminish its code.

3. How would you make the ChameleonCritter drop flowers in its old location when it moves?

We could write as the bug's move method, write the next codes.

```
Public void makeMove(Location loc) {
    Location oldLocation = getLocation();
    setDirection(getLocation().getDirectionToward(loc));
    super.makeMove(loc);
    if (!oldLocation.equals(loc) {
        Flower flower = new Flower(getColor());
        flower.putSelfInGrid(getGrid(), oldLocation());
    }
}
```

4. Why doesn't ChameleonCritter override the getActors method?

It doesn't need to override this method because it share the same list of actors as Critter to processes.

5. Which class contains the getLocation method?

The Actor class.

6. How can a Critter access its own grid?

It can access its own grid by the getGrid method, which inherit from the Actor class

Set 9

1. Why doesn't CrabCritic override the processActors method?

Because the CrabCritic share the same behaviors as the Critter in processActors method, which is eating all of the neighbors.

2. Describe the process a CrabCritic uses to find and eat other actors. Does it always eat all neighboring actors? Explain.

The CrabCritic's getActors method will find the neighbors which is in front, right-front, and left front of it to eat, and other neighboring locations won't, so it doesn't eat all neighboring actors.

3. Why is the getLocationsInDirections method used in CrabCritic?

The method received an array of directions as parameter and return all the valid adjacent locations in the directions. So in CrabCritic, it call the method and give particular directions to determine which location can move.

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

(4,4), (4,3) and (4,5)

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

Similarities:

they both randomly choose their next location in the possible locations list instead of the direction that are moving.

Differences:

A crab critter only move to its left or right when critter can move any possible adjacent neighboring locations. When crab critter can't move, it will turn right or left when critter can't.

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

When the parameter loc in makeMove is equal to the current location.

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

Because CrabCritic inherits the processActors method from the Critter class, and this class can't remove rocks nor critters.