Do You Know?

Set 7

The source code for the Critter class is in the critters directory

1. What methods are implemented in Critter?

Answer: Method act, getActors, processActors, getMoveLocation, selectMoveLocation, makeMove.

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

Answer: They are getActors, processActors, getMoveLocations, selectMoveLocation, makeMove.

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

Answer: Yes. If subclasses of Critter need another location to get actors, it should override the getActors method.

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

Answer: The critter can eat actors which are not rocks or critters.

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

Answer: First, getMoveLocations method. This method return some empty adjacent locations around the current location so that the critter ensures the determination of which locations are candidates for the move. The returned list will pass the method selectMoveLocation.

Second, selectMoveLocation method. This method randomly select a location from which can be moved. If no location can be moved, it return the current location. The returned location will pass the method makeMove.

Thrid, makeMove. This method make the critter move to the given location which is not null.

6. Why is there no Critter constructor?

Answer: Because Critter extends Actor and class Actor has a default constructor. If there no constructor in class, Java will create a default constructor which will call the Actor default constructor automatically.

Do You Know?
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?

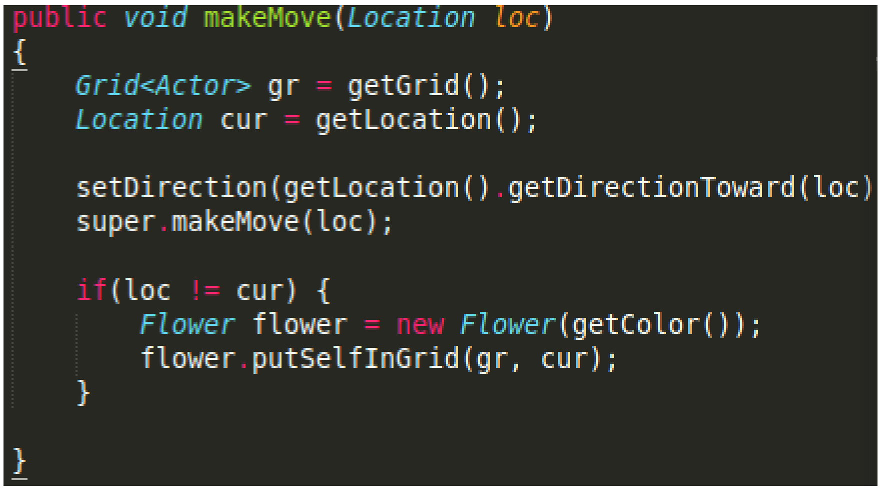
Answer: Because it override the mothod makeMove and getActors. And the method act includes the mothod makeMove and getActors.

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

Answer: The method makeMove of ChameleonCritter first determination a new direction it want to move. Then it call the super.makeMove to move to the new direction really and it behaves as a Critter.

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

Answer: A variable to keep the ChameleonCritter's current location. After moving, put a flower in its old location if the current locaiton is different from the old location. Please the following code.



```
public void makeMove(Location loc)
{
    Grid<Actor> gr = getGrid();
    Location cur = getLocation();

    setDirection(getLocation().getDirectionToward(loc)
    super.makeMove(loc);

    if(loc != cur) {
        Flower flower = new Flower(getColor());
        flower.putSelfInGrid(gr, cur);
    }

}
```

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

Answer: There is no need to do it because ChameleonCritter has the same behavior with Critter in the mothod getActors.

5. Which class contains the getLocation method?

Answer: The class Actor.

6. How can a Critter access its own grid?

Answer: Calling the method getGrid inherited by class Actor.

Do You Know?

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?

Answer: A CrabCritter process that it will eat all its neighbors returned by method getActors which is the same as the base class Critter.

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

Answer: First, method getActors find actors only in front of it or in its left front or right front. Then the method processActors eat the actors returned from method getActors. Other neighboring actors will not be eaten.

3. Why is the getLocationsInDirections method used in CrabCritter?

Answer: This method is to find the adjacent locations with the directions of the given array.

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?

Answer: (4,4), (4,3), (4,5) .

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

Answer: Similarities: Critters and CrabCritters will not change the direction when moving and they both randomly choose one from those locations can be moved as the next location.

Differences: A CrabCritter can only move to its left or right. But a Critter can move any possible location of the eight adjacent neighboring locations. A CrabCritter will choose turn left or right if it cannot move but a Critter will not change its direction.

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

Answer: If the parameter loc of the makeMove is equal with the current location, a CrabCritter will turn instead of moving.

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

Answer: The CrabCritter is inherited from the class Critter which method processActors will not eat Critters. Because CrabCritter is Critter, CrabCritter objects will not eat each other.