22ed Aug, 2015

Name：陈炜健　**Student ID**：**13331018**

Do You Know? - Set 7

1. What methods are implemented in Critter?

   answer:

   - act method
   - getActors method
   - processActors method
   - getMoveLocations method
   - selectMoveLocation method
   - makeMove method

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

   answer:

```
ArrayList<Actor> actors = getActors();
processActors(actors);
ArrayList<Location> moveLocs = getMoveLocations();
Location loc = selectMoveLocation(moveLocs);
makeMove(loc);
```

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

   answer: Yes. Override this method in subclasses to look elsewhere for actors to process.

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

   answer: The critter can remove the actors that are not rock and not critter from the grid.

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

answer:

- getMoveLocations method: Gets a list of possible locations for the next move.
- selectMoveLocation method: Selects randomly the location for the next move.
- makeMove method: Moves this critter to the selected location.

6. Why is there no Critter constructor?

answer: Because the Critter extend the Act which has already had a constructor method. It means that the Critter has a default constructor method which is inherited from the Act.

Do You Know? - Set 8

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

answer: Because the ChameleonCritter override the processActor method and the make move method, which are called in the act method.

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

answer: The ChameleonCritter firstly set direction to the location, then move to the location. So inside the method , it set the direction firstly, then call super.makeMove to move to the location.

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

answer:

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

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

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

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

answer: The ChameleonCritter has the same behavior with the Critter towards the getActors method.

5. Which class contains the getLocation method?

answer: The Actor.

6. How can a Critter access its own grid?

answer: Calls the getGrid method.

Do You Know? - Set 9

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

answer: Because the CrabCritter also eat the actors that are not rock and not critter, which is the same behavior with the Critter.

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

answer: Firstly, a CrabCritter gets the actors in the three locations immediately in front, to its front-right and to its front-left, and then it eat them, not all neighboring actors.

3. Why is the getLocationsInDirections method used in CrabCritter?

answer: This method is to find the valid adjacent locations of this critter in different 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?

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

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

answer:

- Similarities: They eat the actors that are not rocks and critters; They move to a new direction without changing their direction.

- Differences: The CrabCritter only can choose the front, front-right and front-left direction to move, when Critter eight direction.

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

answer: Inside the makeMove method, if the parameter "loc" is equal to current location, the CrabCritter will turns.

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

answer: Because the CrabCritter inherited from the Critter, which will not eat each other. A CrabCritter object is a Critter object, so the CrabCritter objects do not eat each other.