

Set 7

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

1. What methods are implemented in Critter?

act, getActors, getMoveLocations, makeMove, processActors, selectMoveLocation

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

getActors, getMoveLocations, makeMove, processActors, selectMoveLocation

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

Yes if they need. Because different kinds of Critters may have different behaviours.

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

Remove all of them.

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

getMoveLocations, selectMoveLocation, makeMove

getMoveLocations returns an arrayList of locations for the next move.

selectMoveLocation selects the location for the next move.

makeMove moves this critter to the given location loc, or removes this critter from its grid if loc is null.

6. Why is there no Critter constructor?

Because Critter has no member variable to initialize, it just defined the way how the actor acts.

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?

The ChameleonCritter overrides makeMove and processActors and act calls these methods.

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

Because the implementation of this part is not modified, just added.

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

Use a variable to remember the original location, and

"flower.putSelfInGrid(getGrid(), oldLoc)", but it is not a good idea because the world may not know about it.

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

Because the requirement and implementation is the same as Critter.

5. Which class contains the getLocation method?

Actor

6. How can a Critter access its own grid?

Call getGrid.

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 both CrabCritter and Critter eats everything except Critter and rock.

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

First get all neighboring actors in the direction, then process them by eating those that are not Critter or rock, .

No, the neighbors on the side or behind the Crab will not be eaten.

3. Why is the getLocationsInDirections method used in CrabCritter?

Because CrabCritter only eats neighbors in the locations immediately in front, to the right-front, or to the left-front of it.

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?

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

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

Similarities: they choose a random location returned by getMoveLocations

Differences: CrabCritter only moves to left or right, Critter moves to all neighboring locations.

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

When there is no location to move, selectMoveLocation returns the current location, and makeMove randomly selects between left or right to turn.

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

Because the processActors inherited from Critter is so written.