13331183
卢建业

Set 3
Assume the following statements when answering the following questions.

```
Location loc1 = new Location(4, 3);
Location loc2 = new Location(3, 4);
```

1. How would you access the row value for loc1?
   Use this Loc1.getRow() statement.
2. What is the value of b after the following statement is executed?

```
boolean b = loc1.equals(loc2);
```

   False.
3. What is the value of loc3 after the following statement is executed?

```
Location loc3 = loc2.getAdjacentLocation(Location.SOUTH);
```

   (4,4)
4. What is the value of dir after the following statement is executed?

```
int dir = loc1.getDirectionToward(new Location(6, 5));
```

   135
5. How does the getAdjacentLocation method know which adjacent location to return?
   GetAdjacentLocation method received a parameter for the direction of the adjecent ,so compare now location can get the adjacent location.

Set 4
1. How can you obtain a count of the objects in a grid? How can you obtain a count of the empty locations in a bounded grid?
   The getOccupiedLocations()method will return the occupied locations of the grid,so call getOccupiedLocations().size() can return the number of occupied location.
Grid.getNumberRows() * Grid.getNumberCols() will return all the number of the location,so Grid.getNumberRows()*Grid.getNumberCols() - Grid.getOccupiedLocations().size()will get the number of the empty locaion in a bounded grid.
2. How can you check if location (10,10) is in a grid?
   The isValid method could check,use Grid.isValid(new Location(10,10)) statement,if it is valid,will return true, else false.

3. Grid contains method declarations, but no code is supplied in the methods. Why? Where can you find the implementations of these methods?
   Because the Grid is an interface,its method will be implement by another class which implements it.we can find that such as AbstractGrid,BoundedGrid and UnboundedGrid imlements it and implements some these method.
4. All methods that return multiple objects return them in an ArrayList. Do you think it would be a better design to return the objects in an array? Explain your answer.
ArrayList doesn't require to initail the size before filling it when array does,so you may first count the number of occupied locations to size the array,but if the Grid can kept gract of the number of occupied locations, use array may as easy as ArrayList.

Set 5
1. Name three properties of every actor.
   Every actor has color,direction and location properties.

2. When an actor is constructed, what is its direction and color?

The default initail direction is North ,when color is blue.

3. Why do you think that the Actor class was created as a class instead of an interface?

Because the actor has both state and behavior,and interface does't has instance variables and implement method,so Actor was a class instead of interface.

4. Can an actor put itself into a grid twice without first removing itself? Can an actor remove itself from a grid twice? Can an actor be placed into a grid, remove itself,

actor can't put itself into a grid twice without first removing itself,if do this, will throw an IllegalStateException error, can't remove itself from a grid twice,it do this will throw an IllegalStateException error, can e placed into a grid, then remove itself, and then put itself back, if do this, will compile and run without error.

5. How can an actor turn 90 degrees to the right?

Call actor.setDirection(90),the actor will turn to the right.

Set 5

1. Which statement(s) in the canMove method ensures that a bug does not try to move out of its grid?

```
If (!gr.isValid(next))
    return fasle;
```

this statements could ensures that a bug does not try to move out of its grid.

2. Which statement(s) in the canMove method determines that a bug will not walk into a rock?

```
Actor neighbor = gr.get(next);
return (neighbor == null) || (neighbor instanceof Flower);
```

these statements work together could determineds that a bug will not walk into a rock.

3. Which methods of the Grid interface are invoked by the canMove method and why?

IsValid() and get() method of the Grid interface are invoke by the canMove, isValid method could ensure the next location can move to and get method could ensure the next location doesn't has conflict actor.

4. Which method of the Location class is invoked by the canMove method and why?
   GetAdjacentLocation method,it can help to find the next possible location.
5. Which methods inherited from the Actor class are invoked in the canMove method?
   GetLocation(),getDirection() and getGrid method.
6. What happens in the move method when the location immediately in front of the bug is out of the grid?
   The bug will remove itself from the grid when he location immediately in front of the bug is out of the grid
7. Is the variable loc needed in the move method, or could it be avoided by calling getLocation() multiple times?

   Yes,it need,it could stores the bug's old location to insert a flower.

8. Why do you think the flowers that are dropped by a bug have the same color as the bug?

Because the flowers dropped by the bug is the same as the color of the bug.

9. When a bug removes itself from the grid, will it place a flower into its previous location?

When call removeSelfFromGrid method,it won't place a flower into its previous location,because this method is inherite from the Actor class whick don't put a flower in their old location,but when you call the removeSelfFromGrid in Bug move method,it will put a flower.

10. Which statement(s) in the move method places the flower into the grid at the bug's previous location?

```
Flower flower = new Flower(getColor());
flower.putSelfInGrid(gr,loc);
```

11. If a bug needs to turn 180 degrees, how many times should it call the turn method?
Since each turn for 45 degree ,so 180 degrees is 4times.