Part 3

**Do you Know?**

**Set 3**

1. How would you access the row value for loc1?

Answer:Loc1.getRow()

1. What is the value of b after the following statement is executed?

boolean b = loc1.equals(loc2);

Answer:false

1. What is the value of loc3 after the following statement is exectued?

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

Answer:(4,4)

1. What is the value of dir after the following statement is executed?

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

Answer:135(degrees)-Southeast [得到一个方向？]

1. How does the getAdjacentLocation method know which adjacent location to return?

Answer：getAdjacentLocation方法中的参数指示了要查找的相邻邻居的方向，它返回了指南针方向上最接近参数列表中给定方向的相邻位置。

**Do you Know?**

**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?

Answer：我们使用gr作为一个Grid object的实例，gr.getOccupiedLocations().size()

将找到任何类型的grid中占用位置的数量，这就是我们要求的object的数量；

gr.getNumRows()\*gr.getNumCols() - gr.getOccupiedLocations().size() 将会找到bounded grid中空闲的位置的数量。

1. How can you check if location (10,10) is in the grid?

Answer:使用gr.isValid(new Location(10,10))

（方法 isValid 返回true如果传入的位置是一个有效的位置）

1. Grid contains method declarations, but no code is supplied in the methods. Why? Where can you find the implementations of these methods?

Answer：Grid是一个接口。在Java中，接口指定另一个类必须实现的方法。你可以在Abstract Grid 、Bounded Grid和Unbounded Grid类中找到这些接口方法的实现。由于Abstract Grid只实现了Grid接口所需的一些方法，所以它被定义为一种抽象类。Bounded Grid和Unbounded Grid 拓展（extend）了Abstract Grid类并且实现了Grid接口剩余的方法。

1. 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.

Answer；对于一个Grid类的用户，使用[]符号去获得一个元素会稍微比使用不用的方法调用要简单，比如说：locs[j] 和 locs.get(j) 对比起来，ArrayList不需要用户在填充列表之前调整列表的大小。一个数组需要。由于 Bounded Grid不跟踪grid中object的数量，你不得不首先统计被占有的位置的数量去规定数组的大小，然后回到grid中去找到并存储每一个位置。如果Grid跟踪被占有的位置的数量，填充一个数组会变得和填充一个ArrayList一样简单。

**Do you Know?**

**Set 5**

1. Name three properties of every actor.

Answer：actor有color，direction和location属性，同时还有一个对grid的引用。

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

Answer：一个actor初始化的color是blue，初始化的direction是North。

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

Answer：一个actor既有状态也有行为，而一个接口不允许程序员去声明实例变量或实现方法。

1. (a) Can an actor put itself into a grid twice without removing itself? (b) Can an actor remove itself from a grid twice?(c) Can an actor be placed into a grid, remove itself, and then put itself back? Try it out. What happens?

Answer：

(a)不，如果一个actor已经在grid中，它不会再将自己放入到grid中。这个版本的BugRunner.java会编译，但是当它运行的时候会报出一个IllegalStateException异常。

public class BugRunner

{

public static void main(String[] args)

{

ActorWorld world = new ActorWorld();

Bug b = new Bug();

world.add(b);

b.putSelfInGrid(b.getGrid(),b.getLocation());

world.add(new Rock());

world.show();

}

}

(b)不，如果一个actor已经从grid中移除，它不能被再次移除。这个版本的BugRunner.java会编译，但是当它运行的时候会报出一个IllegalStateException异常。

public class BugRunner

{

public static void main(String[] args)

{

ActorWorld world = new ActorWorld();

Bug b = new Bug();

world.add(b);

world.add(new Rock());

world.show();

b.removeSelfFromGrid();

b.removeSelfFromGrid();

}

}

(c)是，一个actor可以被放置在grid中，然后被移除，然后重新放置，尝试这个ActorRunner.java，将会编译并且成功运行。

public class ActorRunner

{

public static void main(String[] args)

{

ActorWorld world = new ActorWorld();

Actor a = new Actor();

world.add(a);

Grid g = a.getGrid(); //must remember the grid for placement back in

//the grid

world.add(new Rock());

world.show();

a.removeSelfFromGrid();

a.putSelfInGrid(g,new Location(5,5)); //must specify a location here

}

}

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

Answer：使用 setDirection 方法，如下展示：

setDirection(getDirection() + Location.RIGHT)或者setDirection(getDirection() + 90);

**Do you Know?**

**Set 6**

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

Answer：if (!gr.isValid(next))

return false; （这个方法会判断下一个位置是否有效）

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

Answer：Actor neighbor = gr.get(next);

return (neighbor == null) || (neighbor instanceof Flower);

这两条语句一起作用来确保bug将只会去往空闲的或者被花占用的位置

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

Answer：isValid 和get。这些方法被用来确定下一个位置是否是有效的位置和确认这个位置是空的或者包含一个actor可以被bug代替。

1. Which method of the Location class is invoked by the canMove method and why?

Answer：getAdjacentLocation方法，这个方法通过输入的当前bug的位置寻找下一个可能的位置。

1. Which methods inherited from the Actor class are invoked by the canMove method?

Answer：getLocation, getDirection, getGrid

1. What happens in the move method when the location immediately in front of the bug is out of the grid?

Answer：bug会将自己从grid中移除

1. Is the variable loc needed in the move method, or could it be avoided by calling getLocation() muliple times?

Answer:是，变量loc被需要在move方法中。变量loc在bug移动之前存储它的位置。它被用在在bug移动到新的位置后在bug的旧位置上插入一个flower。

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

Answer：会更容易去看到哪个bug掉了哪个flower因为bug的颜色和flower的颜色是一致的。

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

Answer：如果只是调用removeSelfFromGrid方法，就不会，这个方法是从Actor类继承的。Actors不会放置flower在他们的就位置上。

当removeSelfFromGrid被调用在Bug move方法中，就会。flower 将会被放置在一个空的位置，下述是move 方法中的部分代码：

if (gr.isValid(next))

moveTo(next);

else

removeSelfFromGrid();

Flower flower = new Flower(getColor());

flower.putSelfInGrid(gr,loc);

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

Answer：Flower flower = new Flower(getColor());

flower.putSelfInGrid(gr, loc); //loc是bug的旧位置

1. If a bug needs to turn 180 degrees, how many times should it call the turn method?

Answer：四次，因为每次只能转45度

Part 3 **Group Activity**

1. (a)What will a jumper do if the location in front of it is empty, but the location two cells in front contains a flower or a rock?

Answer:会往45度方向转弯。

(b)What will a jumper do if the location two cells in front of the jumper is out of the grid?

Answer:会往45度方向转弯。

(c)What will a jumper do if it is facing an edge of the grid?

Answer:会往45度方向转弯。

(d)What will a jumper do if another actor (not a flower or a rock) is in the cell that is two cells in front of the jumper?

Answer:会转弯。

(e)What will a jumper do if it encounters another jumper in its path?

Answer:会转弯。

(f)f.Are there any other tests the jumper needs to make?

Answer:还要讨论jumper前面的位置被占有的情况，以及jumper是否能够跳过flower或者rock。

1. (a)Which class should Jumper extend?

Answer:Jumper应该有Actor拓展而来，因为jumper也是一种actor。

(b)Is there an existing class that is similar to the Jumper class?

Answer：Bug类是Jumper类一个很好的模板，canMove和move方法可以被用来写canJump和jump。

(c)Should there be a constructor? If yes, what parameters should be specified for the constructor?

Answer：这取决于我们是否想要jumper的默认颜色和actor的默认颜色不同，或者提供一种方式让用户去构造jumper的颜色。

(d)Which methods should be overridden?

Answer：act方法，jumper的移动方法和其他actor不相同。

(e)What methods, if any, should be added?

Answer：canJump、jump方法，同时增加一个turn方法帮助jumper转弯。

(f)f.What is the plan for testing the class?

Answer：放置jumper并设置以下几种环境：

1. 在grid的边缘，面对边缘
2. 被rocks环绕
3. 被flowers环绕
4. 被Actors环绕
5. 放置一个rock，flower或actor在jumper前面的两个位置