Step2

1. What is the role of the instance variable sideLength?

In this instance, the trace that a BoxBug moves is a square, so the sideLength means the length of the edge of the square.

2. What is the role of the instance variable steps?

To calculates the step that a BoxBug has moved around a edge of the square.

3. Why is the turn method called twice when steps becomes equal to sideLength?

Because a BoxBug has to move around a square, so it needs to turn 90 degree on each corner of the square, which means it needs to called the turn method twice.

4. Why can the move method be called in the BoxBug class when there is no move method in the BoxBug code?

Because the BoxBug has extended the class Bug, and the move method is in the class Bug.

5. After a BoxBug is constructed, will the size of its square pattern always be the same? Why or why not?

Yes. Because BoxBug has no method to change the size of its square pattern.

6. Can the path a BoxBug travels ever change? Why or why not?

Yes. When a BoxBug moves to the edge of the grid, it will change its direction, which may leads to the change of the path a BoxBug travels.

7. When will the value of steps be zero?

When a BoxBug turns, its steps will be zero.

Exercises

5. Study the code for the BoxBugRunner class. Summarize the steps you would use to add another BoxBug actor to the grid.

Because the BoxBug constructor has an int parameter, when we add an BoxBug actor into the grid, we need toinput a parameter; the steps will be following: 1.Click the mouse within the grid, and choose BoxBug(int); then a window dialog will be displayed to asked for the parameter; 2.Input an int number in the dialog, and click “OK”, and a BoxBug actor will be added into the grid.