part2.

do you know:

1. The sideLength is the length of the square the bug make. In other words, the sideLength defines the number of steps that the BoxBug bug can move on each side of the box.
2. The steps records how many steps a BoxBug bug has move on each side of the box.
3. When the BoxBug bug has move sideLight steps, the bug need to turn 90 degrees to start the next side. But the turn method can only has 45 degrees turn, so we should call the turn method twice.
4. The BoxBug class is extends the Bug class, and the Bug class has a method move. So BoxBug inherits the move method from Bug.
5. Yes. The side length is assigned in the constructors function. When BoxBug is constructed, the side length is determined.
6. Yes. If there is a Rock, Bug, Actor or Border in front of the bug, it will turn and start a new path.
7. When the BoxBug is constructed, steps is assigned to 0. And when steps is equal to sigeLength, it should also assigned to 0.

exercise:

1. The path of the CircleBug is an regular octagon. Because the bug turn 45 degrees in each side, it will finally get an regular octagon.
2. The answer is in the code SpiralBug.java.
3. The answer is in the code ZBug.java.
4. The answer is in the code DancingBug.java.
5.

    1). Create a BoxBug object with the desired length.The code is as follows:

    ```
    BoxBug aboxbugobject = new BoxBug(length);
    ```

    2). Add this object to the grid. The code is as follows:

    ```
    world.add(new Location(a, b), aboxbugobject);
    ```

    If you do not want to add the location, you can use the form as follows:

    ```
    world.add(aboxbugobject);
    ```