**In Depth on the Rabbit class.** Answer these questions to understand how rabbits work.

(1). First, let’s use the code to find all the ways a rabbit can die. Search in Rabbit.java for “alive = false”. Write down each line number this happens on, what *method* that line number is inside of, and what the reason is for the rabbit dying.

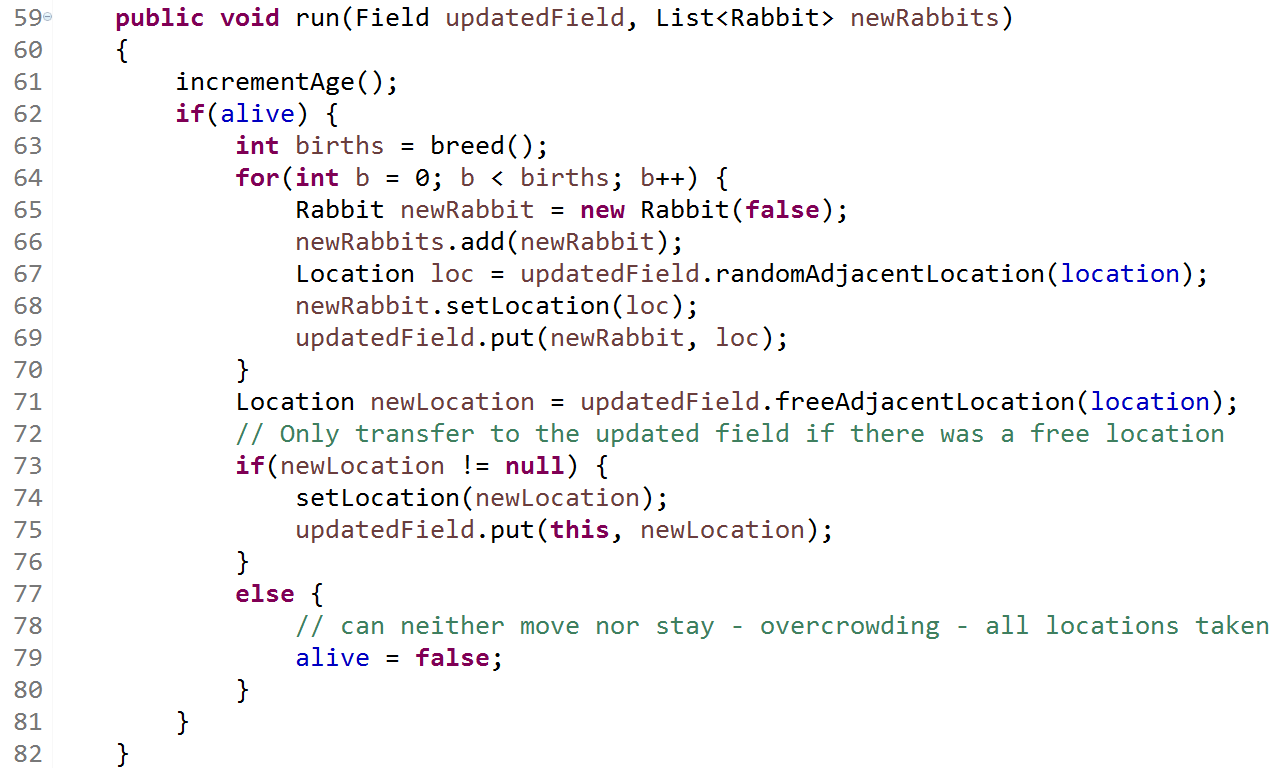
|  |  |  |
| --- | --- | --- |
| Line # | Method name | Reason for dying |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

(2). The **run** method is what has a rabbit move and create babies.

The two inputs are **updatedField** and **newRabbits**.

updatedField: This is where the babies are placed. It represents the field for the next time-step.

newRabbits: This is an ArrayList to hold all the baby rabbits that get created inside this method.

(a). For each line, say out loud to your neighbor what you think it does, and put a **checkmark** next to the line if you know what it does, or a **??** if you’re not sure.

(b). You might think the breed() method (run on line 63) actually creates the babies. It doesn’t. Where do the babies actually get created? What does line 63 actually do?

(c). To place a baby rabbit on the field, you need to do two things. Tell the **rabbit** object where its field location is. And then tell the **field** where the rabbit is. What line of code tells the rabbit its location? \_\_\_\_ What line of code tells the field where the rabbit is? \_\_\_\_\_

(d). Lines 71-81 are about moving the parent rabbit. Line 74 tells the parent rabbit its new location and line 75 tells the field where the rabbit is. You see the variable **newLocation** being used on those lines. On what line number does **newLocation** get its value? \_\_\_\_\_\_\_\_\_ . Put a **circle** around the method that generates the new location.

(e). If line 79 runs, what is true about the **newLocation** variable?