AUFGABE:

On a 15x15 square you'll find a beetle on **each square** initially. When a bird approaches all beetles panic and jump to a random adjacent square.
Beetles sitting on an edge square won't jump off the checkerboard.

The bird approaches 100 times as he's really hungry.

Show us the board after the 25., 50. and 100. approach with
- the number of beetles per square
- the average number of beetles per occupied square
- the square(s) with the highest beetle population

There are no rules how to solve the task and a simple solution (text files, printing to console…) is sufficient.
Please use Java 17.
Please send us the code with a description.
Please also let us know how long it took to solve the task, it shouldn’t take more than a few hours.

Algorithm:

1. Create an a 2 dimensional Array of size (15x15)
2. Initialise the array with 1 in each square
3. For each approach of the bird:
   (i) For each square, find all the possible adjacent coordinates of each beetle.
   Important to know: The beetles may not jump all in the same direction.
   So, each beetle has its random direction.
   (ii) Choose a random cell for each beetle and add the population with 1 to the corresponding adjacent square.
   (iii) Remove all the beetles in the actual cell.
4. At the 25th, 50th, or 100th approach print the board.

N.B: the average number of beetles per occupied square = totalNumberOfBeetles / numberOfOccupiedSquares

Board Initialisation:

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Illustration of possible adjacent where a beetle can jump.

0,0	0,1	0,2	1	1	1	1	1	1	1	1	1	1	1	1
1,0	1	1,2	1	1	1	1	1	1	1	1	1	1	1	1
2,0	2,1	2,2	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	7,6	7,7	7,8	1	1	1	1	1	1
1	1	1	1	1	1	8,6	1	8,8	1	1	1	1	1	1
1	1	1	1	1	1	9,6	9,7	9,8	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	12,12	12,13	12,14
1	1	1	1	1	1	1	1	1	1	1	1	12,12	1	13,14
1	1	1	1	1	1	1	1	1	1	1	1	14,12	14,13	14,14