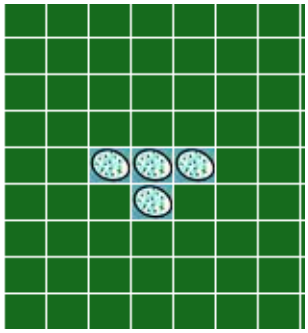
APCompSci **Game of Life Project**
Mr. Laf

An article by Martin Gardner in the October 1970 issue of Scientific American

(see http://www.ibiblio.org/lifepatterns/october1970.html)

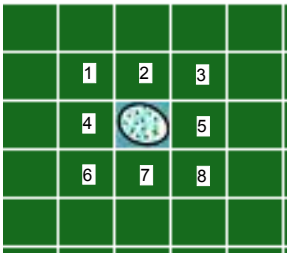
introduced a "game" invented by J.H. Conway of Cambridge University. The game takes place on a rectangular board composed of unit squares. The game models certain simple genetic laws for birth, death and survival.

To see how the game works, click on the `Life.jar` file which will start the game.You will see the green board in which the cells live or die. Go to the middle of the board and click on 4 squares in the following pattern:



The initial "colony" is now populated with 4 cells. Set the **Speed** slider to the left so the process is slowed down. Click the **Start** button and watch what happens on the board in successive generations. Notice how the colony changes in successive generations until it reaches a steady state where there is no change.

Each square on the board can be empty or it can be filled in with a single cell organism. Each square (except border squares) has eight neighbor locations--those that share a common edge or a common corner. The organism "cell" is shown below with all eight numbered neighbor positions:




Starting with an initial colony on the grid, subsequent generations are determined according to the following rules:

(1) birth : an organism will be born in each empty cell that has exactly three neighbors
(2) death : an organism with four or more organisms as neighbors will die from overcrowding;
an organism with fewer than two neighbors will die from loneliness and isolation
(3) survival : an organism with two or three neighbors will survive in the next generation.

All births and death occur simultaneously, and the application of the above laws to an entire board configuration is called a generation. The game is played until one of three events occurs: the colony dies out completely; it reaches a steady state; or a specified number of maximum generations is reached. However, our game version will display subsequent generations until the user clicks **Reset**.

See the following websites to get a further understanding of the Game of Life:

https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life
http://www.math.com/students/wonders/life/life.html
http://www.bitstorm.org/gameoflife/

In this project you are to complete the following methods in the `LifeBoard.java` class. All other classes are supplied in completed form and need not be altered.

(1) Complete the constructor for the `LifeBoard` class. The section of code that needs to be completed initializes the two matrices, temp and tempGrid, to false. (if an element of the matrix is set to true, it indicates the cell is occupied; if set to false, it indicates the cell is empty/unoccupied.)

(2) De-bug the integer method `neighborCount`, which has two parameters specifying the row and column of the matrix element whose neighbors are to be counted. The method has been written; however, there are some bugs which need to be repaired.

(3) Complete the boolean method `isAnyCellAlive`, which returns true if at least one cell in grid is alive; otherwise, it returns false.

(4) Complete the void method `clear` which sets all the elements of grid and tempGrid matrix to false. Afterwards, it repaints the board.

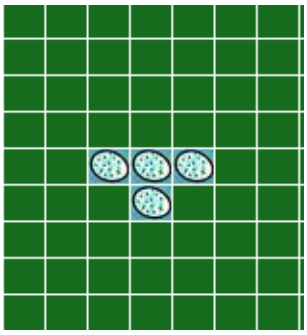
(5) Complete the method `makeNextGeneration` which defines the grid for a subsequent generation based upon Conway's rules (1) through (3) (above) in the Game of Life. `makeNewGeneration` implements the following algorithm:

```
loop through each matrix element in grid matrix
 {
   count the neighbors for each cell (count) of grid
   if (count = 3) and grid[row][col] is empty
     cell is born (set to true) in next generation (tempGrid)
   else if (count = 2 or count = 3) and grid cell is occupied
     cell survives in next generation (tempGrid)
   else
```

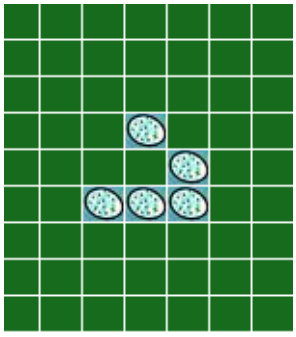
```
        cell dies in next generation (tempGrid)
    }
copy contents of new board(tempGrid) into old board (grid)
repaint
```

When you have completed the above 5 steps, test your program with the following initial configurations:

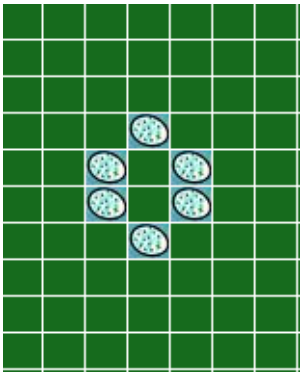
(i)



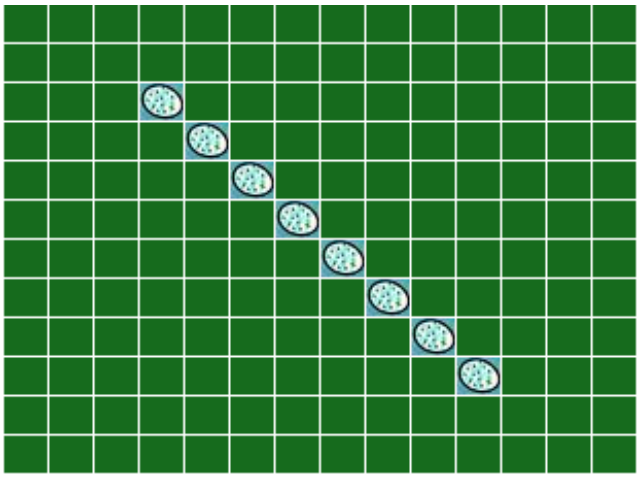
(ii) glider



(iii) behive



(iv)



Using some designs from the web sites listed above, test three more configurations. Conway originally conjectured that no pattern can grow indefinitely—i.e., that for any initial configuration with a finite number of living cells, the population cannot grow beyond some finite upper limit. He was proven wrong the very next month after his challenge appeared in October, 1970.

Test one of the configurations (see https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life) which cause infinite growth to see what happens.