SESSION 5: MATRIXES (TWO-DIMENSIONAL ARRAYS)

GOAL:

- Practice with matrixes (two-dimensional arrays).

EXERCISE:

Version 1:
Implement a program that defines the board for a minesweeper game with the following characteristics:

- The board will be represented by a matrix (2D array) of NxM integers. The dimensions are to be entered from the keyboard, with the smallest matrix that can be created being 2x2.
- The program will mark X positions of the matrix with the value 1 (bomb), obtaining these positions randomly. The value of X shall be entered by the user. The value of X has to be between 1 and (N*M)/2. The rest of the squares will initially contain the value 0 to indicate that they are available.
- In case of wrongly entered values, the program will indicate that there is an error and will ask for the value again.
- When assigning the squares with a bomb, it must be checked that the randomly selected position does not already have a bomb assigned to it. If it already has one, the program should try a different position until one that is without a bomb is obtained.
- Once the bombs have been randomly assigned to the squares, the status of the board will be displayed.

Note: the Random class will be used to generate random numbers.
Example of initialization and use:

```
import java.util.Random;
…
final static Random RANDOM = new Random();
…
int f = RANDOM.nextInt(10);  // generate integer between 0 & 9
```

Version 2:
Starting from version 1, add the following features:

- Create the board and assign bombs randomly, but do not show the state of the board.
- The game is then started, which will end if the user uncovers a square with a bomb or if the user manages to uncover all the squares that do not have a bomb. To do this, the user will select a position on the board (it must be a valid position), and the program will check the selected square:
    - If it corresponds to a square with a bomb (1), this will be reported, the status of the board will be displayed and the program will terminate.
    - If it is an available square (with a 0), it will be marked with a 2. The program will ask for another position on the board as long as there are available squares. If

there are no available squares left, a congratulations message and the final state of the board will be displayed.

- If it is a previously selected square (value 2), this case will be indicated to the user and the game will continue.

Version 3:

Starting from version 2, add the following features:

- If the selected square is available, in addition to what is indicated for version 2 (mark it with a 2), all the squares immediately around it will be examined to count the bombs (value 1) in them, then informing the user about that number of bombs around the selected square.

MILESTONE 2: SUBMISSION OF ASSIGNMENTS 4 AND 5: 27 November

EXAMPLE OF EXECUTION (FOR VERSION 1):

```
Introduce the number of rows for the board (greater than or equal to 2): 3
Introduce the number of columns for the board (greater than or equal to 2): 4
Introduce the amount of bombs (between 1 and 6): 5

Board:

1 0 1 0
0 1 0 0
0 1 1 0

End of the program
```

EXAMPLE OF EXECUTION (FOR VERSION 2):

```
Introduce the number of rows for the board (greater than or equal to 2): 3
Introduce the number of columns for the board (greater than or equal to 2): 4
Introduce the amount of bombs (between 1 and 6): 5

THE GAME STARTS:

Choose a square:
Introduce the row, value in [0, 2]: 0
Introduce the column, value in [0, 3]: 0
The square is available.

Choose a square:
Introduce the row, value in [0, 2]: 1
Introduce the column, value in [0, 3]: 2
The square is available.

Choose a square:
Introduce the row, value in [0, 2]: 0
```

```
Introduce the column, value in [0, 3]: 3
There is a bomb in the selected square.


Board:

2 1 0 1
0 1 2 0
0 1 1 0

End of the program
```

EXAMPLE OF EXECUTION (FOR VERSION 3):

```
Introduce the number of rows for the board (greater than or equal to 2): 3
Introduce the number of columns for the board (greater than or equal to 2): 4
Introduce the amount of bombs (between 1 and 6): 5

THE GAME STARTS:

Choose a square:
Introduce the row, value in [0, 2]: 1
Introduce the column, value in [0, 3]: 1
The square is available.
The selected square is surrounded by 5 bombs.

Choose a square:
Introduce the row, value in [0, 2]: 2
Introduce the column, value in [0, 3]: 3
The square is available.
The selected square is surrounded by 2 bombs.

Choose a square:
Introduce the row, value in [0, 2]: 0
Introduce the column, value in [0, 3]: 2
The square is available.
The selected square is surrounded by 2 bombs.

Choose a square:
Introduce the row, value in [0, 2]: 0
Introduce the column, value in [0, 3]: 1
There is a bomb in the selected square.

Board:

0 1 2 0
1 2 1 0
0 1 1 2

End of the program
```