ST Assignment 2 Part 2

U18CO021: SAHIL BONDRE

Minesweeper: Write a program that takes 3 arguments from user M, N, and p and produces an M-by-N boolean array where each entry is occupied with probability p. In the minesweeper game, occupied cells represent bombs and empty cells represent safe cells. Print out the array using an asterisk for bombs and a period for safe cells. Then, replace each safe square with the number of neighboring bombs (above, below, left, right, or diagonal) and print out the solution.

Code:

```
package lab_2.q_02;

import java.util.Scanner;

public class Main {
    static boolean isBomb(double p) {
        double x = Math.random();
        return x < p;
    }

    static int findNeighbouringBombs(boolean[][] m, int i, int j) {
        return (m[i + 1][j] ? 1 : 0) +
                (m[i - 1][j] ? 1 : 0) +
                (m[i][j + 1] ? 1 : 0) +
                (m[i][j - 1] ? 1 : 0) +
                (m[i - 1][j - 1] ? 1 : 0) +
                (m[i - 1][j + 1] ? 1 : 0) +
                (m[i + 1][j - 1] ? 1 : 0) +
                (m[i + 1][j + 1] ? 1 : 0);
    }
```

```
    public static void main(String[] args) {
        var sc = new Scanner(System.in);
        System.out.printf("Enter m n and p: ");
        int m = sc.nextInt();
        int n = sc.nextInt();
        double p = sc.nextDouble();

        if (p > 1 || p < 0) {
            System.out.printf("Error: Probability should be between 0
and 1");
            return;
        }


        boolean[][] containsMine = new boolean[m + 2][n + 2];

        System.out.println("Generated Configuration:");
        for (int i = 1; i <= m; ++i) {
            for (int j = 1; j <= n; ++j) {
                containsMine[i][j] = isBomb(p);
                if (containsMine[i][j]) {
                    System.out.print("* ");
                } else {
                    System.out.print(". ");
                }
            }
            System.out.println();
        }

        System.out.println("Solved:");
        for (int i = 1; i <= m; ++i) {
            for (int j = 1; j <= n; ++j) {
                if (containsMine[i][j]) {
                    System.out.print("* ");
                } else {
                    System.out.printf("%d ",
findNeighbouringBombs(containsMine, i, j));
                }
            }
            System.out.println();
        }

    }
}
```

Output:

```
Enter m n and p: 8 8 0.2
Generated Configuration:
. . . . * * * .
* . * * . . * .
* * . . . . . .
. . . . * . . *
. * . . . * . .
. . * . . . . *
* . . . . . . .
. . . . . . * .
Solved:
1 2 2 3 * * * 2
* 4 * * 3 4 * 2
* * 3 3 2 2 2 2
3 3 2 1 * 2 2 *
1 * 2 2 2 * 3 2
2 3 * 1 1 1 2 *
* 2 1 1 0 1 2 2
1 1 0 0 0 1 * 1
```