

```

/*
  Caça-minas. Escreva um programa que tome três argumentos de linha de comando m, n e p e produza uma matriz booleana mxn
  onde cada elemento da matriz está ocupado com uma probabilidade p. No jogo de caça-minas, as células ocupadas representam
  bombas e as células vazias representam células seguras. Imprima a matriz usando um asterisco para bombas e um ponto para
  células seguras. Em seguida, crie um array bidimensional inteiro com o número de bombas vizinhas (acima, abaixo, esquerda,
  direita ou diagonal). Por exemplo:

  * * . . .   * * 1 0 0
  . . . . .   3 3 2 0 0
  . * . . .   1 * 1 0 0

*/

public class Minas {

    // gera um mapa mxn com probabilidade p para as bombas
    public static boolean[][] geraMapa(int m, int n, double p) {
        boolean mapa[][] = new boolean[m][n];
        for (boolean[] linha : mapa) // forma mista (laço externo sem índice, interno com)
            for (int i = 0; i < linha.length; i++) {
                if (Math.random() < p) // aloca bomba com probabilidade p
                    linha[i] = true;
            }
        return mapa;
    }

    // retorna true se tem mina em minas[lin][col]
    public static boolean temMina(boolean[][] minas, int lin, int col) {
        return (lin >= 0) && (lin < minas.length) && (col >= 0)
            && (col < minas[0].length && minas[lin][col]); // a ordem importa!
    }

    // imprime as duas matrizes
    public static void imprime(boolean[][] mapa, int[][] vizinhos) {
        for (int i=0; i<mapa.length; i++) {
            for (int j=0; j<mapa[0].length; j++)
                System.out.printf("%s ", mapa[i][j]?"*":".");
            System.out.print("\n");
            for (int j=0; j<mapa[0].length; j++)
                System.out.printf("%s ", vizinhos[i][j]=='*'?"" : vizinhos[i][j]);
            System.out.println();
        }
    }

    // completa a matriz de vizinhos a partir da matriz de booleana
    public static int[][] geraMapaVizinhos(boolean[][] mapa) {
        int[][] vizinhos = new int[mapa.length][mapa[0].length];
        for (int i = 0; i < mapa.length; i++)
            for (int j = 0; j < mapa[0].length; j++)
                if (mapa[i][j])
                    vizinhos[i][j] = '*';
                else {
                    int count = 0;
                    /*
                     Contar as minas nas 8 células adjacentes

                     NO  N  NE
                      \  |  /
                     0---Celula---L
                      /  |  \
                     SO  S  SE

                     Celula-->corrente   (i , j )
                     N --> Norte          (i-1, j )
                     S --> Sul            (i+1, j )
                     L --> Leste           (i , j+1)
                     O --> Oeste           (i , j-1)
                     NE --> Nordeste       (i-1, j+1)
                     NO --> Noroeste       (i-1, j-1)
                     SE --> Sudeste        (i+1, j+1)
                     SO --> Sudoeste       (i+1, j-1)
                    */

                    // Testando células vizinhas
                    // -----(Norte) -----
                    if (temMina(mapa, i - 1, j))
                        count++;

                    // -----(Sul)-----
                    if (temMina(mapa, i + 1, j))
                        count++;

                    // -----(Leste)-----
                    if (temMina(mapa, i, j + 1))
                        count++;
                }
    }
}

```

```
// -----(Oeste)-----
if (temMina(mapa, i, j - 1))
    count++;

// -----(Nordeste)-----
if (temMina(mapa, i - 1, j + 1))
    count++;

// -----(Noroeste)-----
if (temMina(mapa, i - 1, j - 1))
    count++;

// -----(Sudeste)-----
if (temMina(mapa, i + 1, j + 1))
    count++;

// -----(Sudoeste)-----
if (temMina(mapa, i + 1, j - 1))
    count++;
vizinhos[i][j] = count;
    }
    return vizinhos;
}

//toma 3 argumentos da linha de comando m,n e p
public static void main(String[] args) {
    int linhas = Integer.parseInt(args[0]); // leitura de m
    int colunas = Integer.parseInt(args[1]); // leitura de n
    double prob = Double.parseDouble(args[2]); // probabilidade [0,1)
    boolean[][] minas = geraMapa(linhas, colunas, prob);
    int[][] vizinhos = geraMapaVizinhos(minas);
    imprime(minas, vizinhos);
}
}
```