```
// mineSweeper.java (class Block (part of the MVC model))
// Copyright Dave Binkley 2018
/**
 * This is (kind of)my code! Its goal is to hold a game board of
 * Block objects and alter those Blocks
 * CS 312 - Assignment 8
 * @author Andrew Fallon (inspired by the analysis and design of Dave Binkley)
 * @version 1.0 11/23/18
 */

abstract class Block
{
    protected boolean markedAsMine;

    protected Block()
    {
        markedAsMine = false;
    }

    protected void markAsMine()
    {
        markedAsMine = true;
    }

    protected abstract boolean correctlyGuessed();
    protected abstract void guessSafe();
    protected abstract void tellNeighborsAboutMine(GameBoard gb, int r, int c);
    protected abstract void incrementMineCount();
    protected abstract String displayAs();
}


class MineBlock extends Block   // [ no instance variables ]
{
    public void guessSafe()
    {
        System.out.println("BOOM");
        System.exit(0);
    }

    public void tellNeighborsAboutMine(GameBoard gb, int r, int c)
    {
        gb.incrementCountForSurroundingBlocks(r, c);
    }

    public void incrementMineCount() {}

    public boolean correctlyGuessed()
    {
        return markedAsMine;
    }

    public String displayAs()
    {
        if(markedAsMine)
            return "M ";
        else
            return ". ";
    }
}
```



```
class NumberBlock extends Block
{
    private int adjacentMineCount;
    private boolean exposed;

    public NumberBlock()
    {
        adjacentMineCount = 0;
        exposed = false;
    }

    public void guessSafe()
    {
        exposed = true;
        markedAsMine = false;
    }

    public void incrementMineCount()
    {
        adjacentMineCount++;
    }

    public void tellNeighborsAboutMine(GameBoard gb, int r, int c){}

    public boolean correctlyGuessed()
    {
        return !markedAsMine;
    }

    public String displayAs()
    {
        if(markedAsMine)
            return "M ";
        if(exposed)
            return adjacentMineCount + " ";
        else
            return ". ";
    }
}
```

```
/**
 * This is my code! Its goal is to hold a game board of
 * Block objects and alter those Blocks
 * CS 312 - Assignment 8
 * @author Andrew Fallon (inspired by the analysis and design of Dave Binkley)
 * @version 1.0 11/23/18
 */
// mineSweeper.java (class GameBoard (part of the MVC model))
// Copyright Dave Binkley 2018
import java.util.Random;

public class GameBoard
{
    protected Block[][] grid;
    protected View view;
    public static final int BOARD_SIZE = 5;
    public static final int NUMBER_OF_MINES = 3;

    public GameBoard(View v)
    {
        grid = new Block[BOARD_SIZE][BOARD_SIZE];
        view = v;
        int minesLeftToAdd = NUMBER_OF_MINES;
        for(int r=0;r<BOARD_SIZE;r++)
        {
            for(int c=0;c<BOARD_SIZE;c++)
            {
                if(minesLeftToAdd > 0)
                {
                    Block mineBlock = new MineBlock();
                    grid[r][c] = mineBlock;
                    minesLeftToAdd--;
                }
                else
                {
                    Block numBlock = new NumberBlock();
                    grid[r][c] = numBlock;

                }
            }
        }

        Random rand = new Random();
        int mineCount = NUMBER_OF_MINES;

        for(int r=0; r<BOARD_SIZE; r++)
        {
            for(int c=0; c<BOARD_SIZE; c++)
            {
                if(mineCount>0)
                {
                    Block temp = grid[r][c];
                    int _r = rand.nextInt(BOARD_SIZE);
                    int _c = rand.nextInt(BOARD_SIZE);
                    grid[r][c] = grid[_r][_c];
                    grid[_r][_c] = temp;
                }
            }
        }

        for(int r=0; r<BOARD_SIZE; r++)
            for(int c=0; c<BOARD_SIZE; c++)
              grid[r][c].tellNeighborsAboutMine(this, r, c);
    }

    public void guessBlockIsSafe(int r, int c)
    {
        grid[r][c].guessSafe();
    }
```



```
    public void incrementCountForSurroundingBlocks(int r, int c)
    {
        for(int i=r-1; i<=r+1; i++)
        {
            for(int j=c-1; j<=c+1; j++)
            {
                if(onBoard(i, j))
                {
                    grid[i][j].incrementMineCount();
                }
            }
        }
    }

    protected boolean onBoard(int r, int c)
    {
        return (r>=0 && r<BOARD_SIZE && c>=0 && c<BOARD_SIZE);
    }

    public void markBlockAsMine(int r, int c)
    {
        grid[r][c].markAsMine();
    }

    public boolean minesAllFound()
    {
        for(int r=0; r<BOARD_SIZE; r++)
            for(int c=0; c<BOARD_SIZE; c++)
                if(!grid[r][c].correctlyGuessed())
                    return false;
        return true;
    }

    public String displayAs(int r, int c)
    {
        return grid[r][c].displayAs();
    }
}
```