```
// mineSweeper.java (class Block (part of the MVC model))
// Copyright Dave Binkley 2018

/**
* This is my code! ItâM-^@M-^Ys goal is to be Minesweeper
* CS 312 - Assignment 8
* @author Liam Creamer
* @version 1.0 11/28/2018
*/

abstract class Block
{
    protected boolean markedAsMine;

    public abstract String cheat();

    public abstract void incrementAdjacentMineCount();
    public abstract String displayAs();
    public abstract boolean correctlyGuessed();
    public abstract void tellNeighborsAboutMine(GameBoard gb, int row, int col);
    public abstract void guessSafe();

    public void initialize()
    {
        markedAsMine = false;
    }

    public void markAsMine()
    {
        markedAsMine = true;
    }

}


class MineBlock extends Block   // [ no instance variables ]
{
    public MineBlock()
    {
        this.initialize();
    }

    public String cheat()
    {
        return "M";
    }

    public void incrementAdjacentMineCount()
    {

    }

    public void guessSafe()
    {
        System.out.println("Boom");
        System.exit(0);
    }

    public void tellNeighborsAboutMine(GameBoard gb, int row, int col )
    {
        gb.incrementCountForSurroundingBlocks(row, col);
    }

    public boolean correctlyGuessed()
    {
        return markedAsMine;
    }

    public String displayAs()
```

```
    {
        if(markedAsMine)
        {
            return "M";
        }
        return ".";
    }

}


class NumberBlock extends Block
{
    protected int adjacentMineCount;
    protected boolean exposed;

    public NumberBlock()
    {
        this.initialize();
    }

    public String cheat() { return ""+"0123456789".charAt(adjacentMineCount); }

    @Override
    public void initialize()
    {
        adjacentMineCount = 0;
        exposed = false;
    }

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

    public void incrementAdjacentMineCount()
    {
        adjacentMineCount++;
    }

    public boolean correctlyGuessed()
    {
        return !markedAsMine;
    }

    public String displayAs()
    {
        if(markedAsMine)
            return "M";

        if(exposed)
        {
            return "" + adjacentMineCount;
        }
        else
        {
            return ".";
        }
    }

    @Override
    public void tellNeighborsAboutMine(GameBoard gb, int row, int col)
    {

    }
}
```

```
// mineSweeper.java (class GameBoard (part of the MVC model))
// Copyright Dave Binkley 2018

/**
* This is my code! ItâM-^@M-^Ys goal is to be Minesweeper
* CS 312 - Assignment 8
* @author Liam Creamer
* @version 1.0 11/28/2018
*/


import java.util.Random;
public class GameBoard
{
  protected final int BOARD_SIZE = 6;
  protected final int NUMBER_OF_MINES = 6;
  protected final int MAXIMUM = 6;

  protected View view;
  protected Block[][] grid = new Block[BOARD_SIZE][BOARD_SIZE];

  protected void cheat()
  {


    for(int r=0; r<BOARD_SIZE; r++)
    {
      String S = "";
      for(int c=0; c<BOARD_SIZE; c++)
        S += grid[r][c].cheat();
      System.out.println(S);
    }
  }


  public GameBoard(View v)
  {
    view = v;
    int count = 0;
    Random random = new Random();

    for(int r=0; r<BOARD_SIZE; r++)        // fill in mine counters
    {
      for(int c=0; c<BOARD_SIZE; c++)
      {
          if(count< NUMBER_OF_MINES)
          {
              MineBlock mineBlock = new MineBlock();
              grid[r][c] = mineBlock;
              count++;
          }
          else
          {
              NumberBlock numberBlock = new NumberBlock();
              grid[r][c] = numberBlock;
          }
      }
    }

    Block temp;
    int counter = 0;
    for(int j = 0;j < BOARD_SIZE; j++)
    {
       for (int k = 0; k< BOARD_SIZE; k++)
       {
           if(counter < NUMBER_OF_MINES )
           {
               temp = grid[j][k];
               int tempRow = random.nextInt(MAXIMUM);
```

```
               int tempCol = random.nextInt(MAXIMUM);
               grid[j][k] = grid[tempRow][tempCol];
               grid[tempRow][tempCol] = temp;
           }
       }
    }

    for(int e = 0; e < BOARD_SIZE; e++)
       for (int f = 0; f < BOARD_SIZE; f++)
           grid[e][f].tellNeighborsAboutMine(this, e, f);

     //cheat();  // for testing ... and amazing your friends!
  }

  public void guessBlockIsSafe(int row, int col)
  {
      grid[row][col].guessSafe();
  }

  public void markBlockAsMine(int row, int col)
  {
      grid[row][col].markAsMine();
  }

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

  public void incrementCountForSurroundingBlocks(int row, int col)
  {
         if(this.onBoard(row -1, col -1))
             grid[row -1][col -1].incrementAdjacentMineCount();

         if(this.onBoard(row -1, col))
             grid[row -1][col].incrementAdjacentMineCount();

         if(this.onBoard(row-1, col +1))
             grid[row -1][col +1].incrementAdjacentMineCount();

         if(this.onBoard(row, col -1))
             grid[row][col -1].incrementAdjacentMineCount();

         if(this.onBoard(row, col +1))
             grid[row][col +1].incrementAdjacentMineCount();

         if(this.onBoard(row +1, col -1))
             grid[row +1][col -1].incrementAdjacentMineCount();

         if(this.onBoard(row +1, col))
             grid[row +1][col].incrementAdjacentMineCount();

         if(this.onBoard(row +1, col +1))
             grid[row +1][col +1].incrementAdjacentMineCount();
  }



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

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

```
// mineSweeper.java (TTY main)
// Copyright Dave Binkley 2018


public class TTY
{
  static public void main(String[] args)
  {
    int r = 0;
    int c = 0;
    char choice = ’X’;
    View v = new TTYView();
    GameBoard b = new GameBoard(v);
    v.setModel(b);

    v.initialDisplay();

    // the ’controller’ (this code) gets input "events" and updates the
    // model (which requests view updates as required)
    do
    {
      System.out.println("enter guess: ");
      try
      {
        choice = (char) System.in.read(); // [ rather unforgiving IO ]
        r = (char) System.in.read() - ’0’;
        c = (char) System.in.read() - ’0’;
        System.in.read(); // return
      }
      catch(Exception e)
      {
      }

      System.out.println("r = " + r + " c = " + c + " choice " + choice);
      if (choice == ’M’)
        b.markBlockAsMine(r,c);
      else
        b.guessBlockIsSafe(r,c);
    } while (!b.minesAllFound());

    System.out.println("yipee!!  you found them all");
  }
}
```

```
// mineSweeper.java (class TTYView (an MVC view))
// Copyright Dave Binkley 2018


interface View
{
  public void setModel(GameBoard m);
  public void initialDisplay();
  abstract public void update(int r, int c);
  abstract public void gameOver(String msg);
}


// this abstract class provides default versions of the methods
// setModel, initialDisplay, and gameOver
abstract class aView implements View
{
  protected GameBoard model = null;

  public void setModel(GameBoard m) {model = m;}
  public void initialDisplay() {update(0, 0);}
  public void gameOver(String msg) {System.out.println(msg); System.exit(0);}

  abstract public void update(int r, int c);
}


class TTYView extends aView
{
  public void update(int _r, int _c)   // tty version ignores _r and _c
  {
    for(int r = 0; r < model.BOARD_SIZE; r++)
    {
      String S = "";
      for(int c = 0; c < model.BOARD_SIZE; c++)
        S += model.displayAs(r, c);
      System.out.println(S);
    }
  }
}
```