```java
package GameOfLife;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.util.Random;

/**
 * Cell in the Game of Life
 */
public class Cell extends JPanel{

    public static Rule rule = new Rule();
    private boolean living;

    private final Color aliveColor;
    /**
     * @param deadColor color of a cell that is dead
     * @param aliveColor color of a cell that is alive
     */
    public Cell(Color deadColor, Color aliveColor){
        /*
       mouse listener -> so that cell can be turned alive/dead by the user
        */


        MouseListener listener = new MouseAdapter() {
            //state from dead/alive
            public void mousePressed(MouseEvent e){
                living = !living;
                repaint();
            }

           //we can slide our mouse nicely
            public void mouseEntered(MouseEvent e) {
                if(SwingUtilities.isLeftMouseButton(e)){
                    living = !living;
                    repaint();
                }
            }
        };

        this.addMouseListener(listener);
        // set the background of the grid to the dead cell color
        setBackground(deadColor);

        this.aliveColor = aliveColor;

        Random random = new Random();
        living = random.nextBoolean();
    }


    public boolean isCellAlive(int aliveNeighbours){
        if(!living)
        {
            String[] numbers = rule.getRuleForDeadCells().replaceAll(" ", "").split(",");
            for (String s: numbers)
            {
                if(aliveNeighbours == Integer.parseInt(s))
                {
                    return true;
                }
            }
        }else if(living)
        {
            String[] numbers = rule.getRuleForAliveCells().split(",");
            for (String s: numbers)
            {
                if(aliveNeighbours == Integer.parseInt(s))
                {
                    return true;
                }
            }
        }
        return false;
    }


    public void setAlive(boolean alive){
```

```
 81         living = alive;
 82     }
 83     public void setDead(boolean alive){
 84         living = !alive;
 85     }
 86
 87
 88     public boolean isLiving(){
 89         return living;
 90     }
 91
 92
 93     // set fill color to aliveColor painting cells
 94     @Override
 95     public void paintComponent(Graphics g){
 96         super.paintComponent(g);
 97
 98         g.setColor(aliveColor);
 99         if (living) {
100             g.fillRect(0, 0, getWidth() - 1, getHeight() - 1);
101         } else {
102             g.drawRect(0, 0, getWidth() - 1, getHeight() - 1);
103         }
104     }
105
106 }
107
```