

홀짝 게임 과제

실행화면



소스코드

Main.java

```
package selectOddEven;

public class Main {
    public static void main(String[] args) {
        swing s = new swing();
    }
}
```

logic.java

```
package selectOddEven;
```

```
public class logic {  
    public String selectOdd() {  
        return "홀";  
    }  
  
    public String selectEven() {  
        return "짝";  
    }  
  
    public boolean selectAnswer(int rNumber, String select) {  
        if (select.equals("홀")) {  
            if (rNumber % 2 == 1) {  
                return true;  
            }  
            else return false;  
        }  
        else if (select.equals("짝")) {  
            if (rNumber % 2 == 0) {  
                return true;  
            }  
            else return false;  
        }  
        else {  
            System.out.println("Error!");  
            return false;  
        }  
    }  
  
    public int makeRandomNumber () {  
        int rNumber = (int)(Math.random() * 10) + 1;  
        return rNumber;  
    }  
}
```

swing.java

```
package selectOddEven;

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

class swing extends JFrame {
    private CenterPanel CPanel;
    private SouthPanel SPanel;
    private logic gameLogic = new logic();

    swing() {
        super("Multiple Panels and Layouts");

        setTitle("홀짝 게임");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(500, 500);
        //setContentPane(new panel());
        Container c = getContentPane();

        CPanel = new CenterPanel();
        SPanel = new SouthPanel(CPanel, gameLogic);

        c.setLayout(new BorderLayout());
        c.add(CPanel, BorderLayout.CENTER);
        c.add(SPanel, BorderLayout.SOUTH);

        setVisible(true);
    }
}

class CenterPanel extends JPanel {
    public JLabel rNumberLabel = new JLabel("?");
    public JLabel stateLabel = new JLabel("무엇일까요");

    CenterPanel() {
        labelSetting();

        add(rNumberLabel);
        add(stateLabel);
    }

    public void labelSetting() {
        setLayout(null);
    }
}
```

```

        setPreferredSize(new Dimension(500, 410));

        Font font1 = new Font("Gulim", Font.BOLD, 50);
        Font font2 = new Font("Gulim", Font.BOLD, 40);

        rNumberLabel.setFont(font1);
        stateLabel.setFont(font2);
        rNumberLabel.setForeground(Color.YELLOW);
        rNumberLabel.setBounds(235, 135, 70, 70);
        stateLabel.setBounds(150, 300, 300, 50);
    }

    public void paintComponent (Graphics g) {
        super.paintComponent(g);

        g.setColor(Color.magenta);
        g.fillRect(185, 100, 130, 130);
    }
}

class SouthPanel extends JPanel implements ActionListener {
    JButton oddButton = new JButton("홀");
    JButton evenButton = new JButton("짝");
    JButton selectButton = new JButton("확인");
    JButton restartButton = new JButton("다시");

    private CenterPanel cp;
    private logic gameLogic;
    private String labelPrint = "";

    SouthPanel(CenterPanel CPanel, logic gameLogic) {
        this.cp = CPanel;
        this.gameLogic = gameLogic;

        setLayout(null);
        setPreferredSize(new Dimension(500, 90));

        ButtonSetting();
        addListeners();

        add(oddButton);
        add(evenButton);
        add(selectButton);
        add(restartButton);
    }
}

```

```

public void ButtonSetting () {
    oddButton.setBounds(10, 10, 100, 50);
    evenButton.setBounds(130, 10, 100, 50);
    selectButton.setBounds(250, 10, 100, 50);
    restartButton.setBounds(370, 10, 100, 50);
}

private void addListeners() {
    oddButton.addActionListener(this);
    evenButton.addActionListener(this);
    selectButton.addActionListener(this);
    restartButton.addActionListener(this);
}

@Override
public void actionPerformed(ActionEvent e) {
    Object source = e.getSource();

    if (source == oddButton || source == evenButton) {
        clickOddOrEven(source);
    } else if (source == selectButton) {
        clickSelect();
    } else if (source == restartButton) {
        clickRestart();
    }
}

private void clickOddOrEven (Object source) {
    cp.labelSetting();

    if (source == oddButton) {
        labelPrint = gameLogic.selectOdd();
        cp.stateLabel.setText("홀 (선택중)");
    }
    else if (source == evenButton) {
        labelPrint = gameLogic.selectEven();
        cp.stateLabel.setText("짝 (선택중)");
    }
}

private void clickSelect() {
    if (labelPrint.isEmpty()) {
        cp.stateLabel.setBounds(100, 300, 500, 50);
        cp.stateLabel.setText("먼저 선택하세요");
        return;
    }
}

```

```

int rNumber = gameLogic.makeRandomNumber();
cp.rNumberLabel.setText(Integer.toString(rNumber));
cp.stateLabel.setBounds(130, 300, 500, 50);

if (gameLogic.selectAnswer(rNumber, labelPrint)) {
    if (rNumber == 10) {
        cp.rNumberLabel.setBounds(220, 135, 70, 70);
    }

    cp.stateLabel.setText(labelPrint + "! " + "맞았어요.");
} else {
    cp.stateLabel.setText(labelPrint + "! " + "아쉽군요.");
}

oddButton.setEnabled(false);
evenButton.setEnabled(false);
selectButton.setEnabled(false);
}

private void clickRestart() {
    labelPrint = "";
    cp.rNumberLabel.setText("?");
    cp.stateLabel.setText("무엇일까요");
    cp.labelSetting();

    oddButton.setEnabled(true);
    evenButton.setEnabled(true);
    selectButton.setEnabled(true);
}
}

```