

자바 클래스 메소드 문서

- MainFrame 클래스

```
package Phone;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.net.URI;

// 메인 프레임 (전화번호부, 계산기, 게임)
public class MainFrame extends JFrame{

    public MainFrame() {
        Container ct = getContentPane();
        GridLayout gl = new GridLayout(3, 1, 20, 20);
        ct.setLayout(gl);

        JButton jb1 = new JButton("전화번호부");
        JButton jb2 = new JButton("계산기");
        JButton jb3 = new JButton("게임-Up&Down");

        jb1.setActionCommand("WhitepagesClicked");
        jb2.setActionCommand("CalculatorClicked");
        jb3.setActionCommand("GameClicked");

        ActionListener buttonClickHandler = new ButtonClickHandler();

        jb1.addActionListener(buttonClickHandler);
        jb2.addActionListener(buttonClickHandler);
        jb3.addActionListener(buttonClickHandler);

        ct.add(jb1);
        ct.add(jb2);
        ct.add(jb3);

        setTitle("Phone");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(250,300);
        setVisible(true);
        setLocationRelativeTo(null);
    }

    public static void main(String[] args) {
        new MainFrame();
    }

    class ButtonClickHandler implements ActionListener{
        @Override
        public void actionPerformed(ActionEvent e) {
            String command = e.getActionCommand();
            if("WhitepagesClicked".equals(command)) {
                new WhitePages();
            }else if("CalculatorClicked".equals(command)) {
                new Calculator();
            }else if("GameClicked".equals(command)) {
                new Game();
            }
        }
    }
}
```

생성자 MainFrame()

- 생성자에서 세 버튼을 Grid Layout 으로 배열한다.
- 버튼은 누르면 각각 '전화번호부'는 WhitePages Class, '계산기'는 Calculator Class, '게임'은 Game Class 의 각 화면 (컨테이너)를 생성한다.

메소드 main()

- 프로그램의 시작점으로, 실행 시 MainFrame 화면(컨테이너)을 생성한다.

클래스 ButtonClickHandler

- ActionListener 인터페이스를 구현하여 버튼 클릭 이벤트를 처리한다.
- actionPerformed 메소드는 버튼 클릭 이벤트가 발생할 시, 각 버튼의 명령(command)에 따라 다른 동작을 수행한다.

• WhitePages Class (전화번호부)

```
package Phone;

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

class WhitePages extends JFrame {

    private DefaultListModel<String> contacts;
    private JList<String> contactsList;
    private JTextField nameField;
    private JTextField numberField;

    public WhitePages() {

        Container ct = getContentPane();
        ct.setLayout(new FlowLayout());

        // 전체 패널
        JPanel panel = new JPanel();
        panel.setLayout(new BorderLayout(panel, BorderLayout.PAGE_AXIS));

        // 전화번호부 목록
        contacts = new DefaultListModel<>();
        contactsList = new JList<>(contacts);
        JScrollPane scrollPane = new JScrollPane(contactsList);
        panel.add(scrollPane);

        // 이름 입력 필드
        JPanel inputPanel = new JPanel();
        inputPanel.setLayout(new BorderLayout(inputPanel, BorderLayout.LINE_AXIS));

        JLabel nameLabel = new JLabel("이름");
        nameField = new JTextField(10);
        inputPanel.add(nameLabel);
        inputPanel.add(nameField);
        panel.add(inputPanel);

        // 전화번호 입력 필드
        JPanel inputPanel2 = new JPanel();
        inputPanel2.setLayout(new BorderLayout(inputPanel2, BorderLayout.LINE_AXIS));

        JLabel numberLabel = new JLabel("전화번호");
        numberField = new JTextField(10);
        inputPanel2.add(numberLabel);
        inputPanel2.add(numberField);
        panel.add(inputPanel2);

        // 라디오 버튼 - 카테고리 (가족, 친구, 업무, 기타)
        // 수평방향 패널에 라디오 버튼을 다 넣고, 전체 패널에 라디오버튼패널 추가
        JPanel horizontalPanel = new JPanel(new FlowLayout(FlowLayout.LEFT));

        JRadioButton opt1= new JRadioButton("가족");
        JRadioButton opt2= new JRadioButton("친구");
        JRadioButton opt3= new JRadioButton("업무");
        JRadioButton opt4= new JRadioButton("기타");
        ButtonGroup group = new ButtonGroup();
        group.add(opt1); group.add(opt2); group.add(opt3); group.add(opt4);
```

```

        horizontalPanel.add(opt1); horizontalPanel.add(opt2); horizontalPanel.add(opt3);
        horizontalPanel.add(opt4);

        panel.add(horizontalPanel);

        // 추가 버튼
        JButton addButton = new JButton("추가");
        addButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                String name = nameField.getText();
                String number = numberField.getText();

                JRadioButton selectedRadioButton = getSelectedRadioButton(group);
                String relation = (selectedRadioButton != null) ?
selectedRadioButton.getText() : null;

                if (!name.isEmpty() && !number.isEmpty() && relation != null ) {
                    String contact = name + "(" + relation + ") : " + number ;
                    contacts.addElement(contact);
                    nameField.setText("");
                    numberField.setText("");
                    group.clearSelection();
                } else {
                    JOptionPane.showMessageDialog(WhitePages.this, "모든 정보를 입력해주세요.",
"주의", JOptionPane.WARNING_MESSAGE);
                }
            }
        });
        panel.add(addButton);

        // 삭제 버튼
        JButton removeButton = new JButton("삭제");
        removeButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                int selectedIndex = contactsList.getSelectedIndex();
                if (selectedIndex != -1) {
                    contacts.remove(selectedIndex);
                }
            }
        });
        panel.add(removeButton);

        ct.add(panel);
        setTitle("전화번호부");
        pack();
        setVisible(true);
        setLocationRelativeTo(null);

    }

    // 라디오 버튼에서 텍스트 가져오는 메소드
    private static JRadioButton getSelectedRadioButton(ButtonGroup group) {
        Enumeration<AbstractButton> buttons = group.getElements();
        while (buttons.hasMoreElements()) {
            JRadioButton button = (JRadioButton) buttons.nextElement();
            if (button.isSelected()) {
                return button;
            }
        }
        return null;
    }
}

```

필드

contacts: 전화번호부 목록을 관리하기 위한 DefaultListModel<String> 객체

contactsList: 전화번호부 목록을 표시하는 JList<String> 객체

nameField: 이름을 입력하는 JTextField 객체.

numberField: 전화번호를 입력하는 JTextField 객체

생성자 WhitePages()

JPanel 을 사용하여 전체 패널을 생성하고, 전화번호부 목록과 입력 필드, 라디오 버튼, 추가/삭제 버튼을 가진다.

JScrollPane 을 사용하여 전화번호 기록이 많을 시 전화번호부 목록을 스크롤 가능하도록 구현한다.

사용자가 이름, 전화번호를 입력하고, 카테고리(가족, 친구, 업무, 기타)를 선택할 수 있는 라디오 버튼을 구성한다.

"추가" 버튼을 누르면 입력된 이름, 전화번호, 카테고리 정보를 전화번호부 목록에 추가한다. 이때 빈 칸이 있거나 카테고리를 선택하지 않은 경우에는 경고 메시지가 표시된다.

"삭제" 버튼을 누르면 선택된 항목을 전화번호부 목록에서 삭제한다.

메소드 getSelectedRadioButton(ButtonGroup group)

ButtonGroup 에서 선택된 라디오 버튼을 반환하는 메소드이다.

• Calculator Class (계산기)

```
package Phone;

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

class Calculator extends JFrame implements ActionListener {

    private JLabel jl = new JLabel(" ");
    private String currentInput;
    private double firstNumber;
    private double secondNumber;
    private String operation;
    private double result;

    public Calculator() {
        Container ct = getContentPane();
        ct.setLayout(new GridLayout(3,1));

        JPanel p_del = new JPanel(new GridLayout(1,2));
        JPanel p_btn = new JPanel(new GridLayout(4,4));

        JButton jb_del = new JButton("C");
        JButton jb_back = new JButton("BackSpace");

        JButton jb = new JButton("");
        JButton jb0 = new JButton("0");
        JButton jb1 = new JButton("1");
        JButton jb2 = new JButton("2");
        JButton jb3 = new JButton("3");
        JButton jb4 = new JButton("4");
        JButton jb5 = new JButton("5");
        JButton jb6 = new JButton("6");
        JButton jb7 = new JButton("7");
        JButton jb8 = new JButton("8");
        JButton jb9 = new JButton("9");
        JButton jb_plus = new JButton("+");
        JButton jb_minus = new JButton("-");
        JButton jb_mul = new JButton("*");
        JButton jb_div = new JButton("/");
        JButton jb_eq = new JButton("=");

        p_del.add(jb_del);
        p_del.add(jb_back);

        p_btn.add(jb7);      p_btn.add(jb8);      p_btn.add(jb9);
        p_btn.add(jb_plus);  p_btn.add(jb4);      p_btn.add(jb6);
        p_btn.add(jb_minus); p_btn.add(jb3);      p_btn.add(jb1);
        p_btn.add(jb_mul);   p_btn.add(jb2);      p_btn.add(jb5);
        p_btn.add(jb0);      p_btn.add(jb); p_btn.add(jb_eq);      p_btn.add(jb_div);

        ct.add(jl);
        ct.add(p_del);
        ct.add(p_btn);

        jb0.addActionListener(this);
        jb1.addActionListener(this);
        jb2.addActionListener(this);
        jb3.addActionListener(this);
        jb4.addActionListener(this);
        jb5.addActionListener(this);
        jb6.addActionListener(this);
        jb7.addActionListener(this);
```

```

        jb8.addActionListener(this);
        jb9.addActionListener(this);

        // Operation buttons
        jb_plus.addActionListener(this);
        jb_minus.addActionListener(this);
        jb_mul.addActionListener(this);
        jb_div.addActionListener(this);
        jb_eq.addActionListener(this);

        // Other buttons
        jb_del.addActionListener(this);
        jb_back.addActionListener(this);

        setTitle("Calculator");
        setSize(200, 200);
        setVisible(true);
        setLocationRelativeTo(null);
    }

    public void actionPerformed(ActionEvent ae) {
        String command = ae.getActionCommand();

        if (command.equals("C")) { // Clear all values
            jl.setText("");
            firstNumber = 0;
            secondNumber = 0;
            operation = " ";
        } else if (command.equals("BackSpace")) {
            String currentText = jl.getText();
            if (currentText.length() > 0) {
                jl.setText(currentText.substring(0, currentText.length() - 1));
            }
        } else if (command.equals("0") || command.equals("1") || command.equals("2") ||
            command.equals("3") || command.equals("4") || command.equals("5") ||
            command.equals("6") || command.equals("7") || command.equals("8") ||
command.equals("9")) {
            jl.setText(jl.getText() + command);
        } else if (command.equals("+") || command.equals("-") || command.equals("*")
||command.equals("/")) {
            firstNumber = Double.parseDouble(jl.getText());
            operation = command.toString();
            jl.setText("");
        } else if (command.equals("=")) {
            if (!jl.getText().isEmpty()) {
                secondNumber = Double.parseDouble(jl.getText());
                switch (operation) {
                    case "+":
                        result = firstNumber + secondNumber;
                        break;
                    case "-":
                        result = firstNumber - secondNumber;
                        break;
                    case "*":
                        result = firstNumber * secondNumber;
                        break;
                    case "/":
                        if (secondNumber != 0) {
                            result = firstNumber / secondNumber;
                        } else {
                            jl.setText("Error: Division by zero");
                            return; // exit early if dividing by zero
                        }
                        break;
                }
            }

            jl.setText(String.format("%.2f", result)); // Display the result
            operation = " "; // Reset operation for next calculation
        }
    }

```

```
        firstNumber = 0;
        secondNumber = 0;
        result = 0;
    }
}
}
```

필드

jl: 결과를 표시하는 데 사용되는 JLabel 객체

currentInput: 현재 jl 의 입력 값으로 사용되는 문자열

firstNumber, secondNumber: 계산에 사용되는 첫 번째, 두 번째 숫자를 저장하는 변수

operation: 수행할 연산(덧셈, 뺄셈, 곱셈, 나눗셈)을 저장하는 변수

result: 연산 결과를 저장하는 변수

생성자 Calculator()

숫자 및 연산 버튼, 결과를 표시하는 라벨 등을 생성하여 패널에 추가한다.

각 버튼에 대해 ActionListener 를 등록하여 사용자의 입력을 처리한다.

메소드 actionPerformed(ActionEvent ae)

사용자가 버튼을 클릭할 때 수행되는 메소드이다.

클릭된 버튼의 명령을 확인하고, 해당하는 기능(숫자 입력, 연산, 결과 표시, 지우기 등)을 수행한다.

사용자 숫자 버튼을 클릭하면 해당 숫자를 결과 창에 추가하고, 연산자 버튼을 클릭하면 첫 번째 숫자와 연산자 변수에 값을 할당, '=' 버튼을 클릭하면 두 번째 숫자를 할당한 후 설정된 연산을 수행하고 결과를 표시한다. C 는 전체 계산을 지우고, BackSpace 는 JLabel 의 숫자 중 맨 끝 숫자를 지운다.

.

- Game Class (Up & Down 게임)

```
package Phone;

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

public class Game extends JFrame {
    private int randomNumber;
    private JTextField guessField;
    private JTextArea resultArea;
    private JButton submitButton;

    public Game () {

        randomNumber = (int) (Math.random() * 100) + 1;

        setTitle("Up & Down 게임");
        JPanel inputPanel = new JPanel();
        JLabel guessLabel = new JLabel("숫자를 입력하세요 (1~100):");
        guessField = new JTextField(10);
        submitButton = new JButton("확인");

        inputPanel.add(guessLabel);
        inputPanel.add(guessField);
        inputPanel.add(submitButton);

        resultArea = new JTextArea(10, 20);
        resultArea.setEditable(false);
        JScrollPane scrollPane = new JScrollPane(resultArea);

        submitButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                try {
                    int userGuess = Integer.parseInt(guessField.getText());

                    if (userGuess == randomNumber) {
                        resultArea.append("축하합니다! 숫자를 맞췄습니다.\n");
                        guessField.setEnabled(false);
                        submitButton.setEnabled(false);
                    } else if (userGuess < randomNumber) {
                        resultArea.append("Up! 더 큰 숫자를 입력하세요.\n");
                    } else {
                        resultArea.append("Down! 더 작은 숫자를 입력하세요.\n");
                    }
                } catch (NumberFormatException ex) {
                    resultArea.append("유효한 숫자를 입력하세요.\n");
                }
                guessField.setText(""); // 입력 필드 초기화
            }
        });

        add(inputPanel, BorderLayout.NORTH);
        add(scrollPane, BorderLayout.CENTER);

        pack();
        setVisible(true);
        setLocationRelativeTo(null);
    }
}
```

필드

randomNumber: 1 부터 100 사이의 랜덤 숫자를 저장하는 필드

guessField: 사용자가 숫자를 입력하는 JTextField 객체

resultArea: 게임 결과를 보여주는 JTextArea 객체

submitButton: 사용자의 추측을 제출하는 JButton 객체

생성자 Game()

"숫자를 입력하세요" 라벨과 숫자 입력 필드, 확인 버튼을 포함하는 JPanel 을 생성한다.

게임 결과를 표시할 JTextArea 와 스크롤 기능이 있는 JScrollPane 을 생성한다.

랜덤 숫자를 생성하여 randomNumber 에 저장한다.

"확인" 버튼에 대한 이벤트 리스너를 추가하여 사용자가 입력한 숫자와 랜덤 숫자를 비교하고, 결과를 텍스트 영역에 표시한다.

입력 필드를 초기화하고, 사용자가 맞춘 경우에는 입력 필드와 버튼을 비활성화한다.

이 게임은 사용자가 숫자를 입력하여 랜덤으로 생성된 숫자를 맞추는 게임으로, 사용자가 입력한 숫자와 비교하여 "Up", "Down", 혹은 "축하합니다!" 메시지를 텍스트 영역에 표시한다. 만약 사용자가 정답을 맞히면 입력 필드와 버튼을 비활성화하여 게임이 종료되도록 한다.