```java
package midterm_exam;

import java.util.*;

interface Dic {
    public String get();
    public void set(String x, String y);
    public void print();
}

class KorKor implements Dic {
    String word;
    String meaning;

    KorKor(String word, String meaning) {
        this.word = word;
        this.meaning = meaning;
    }

    @Override
    public String get() {
        return this.meaning;
    }

    @Override
    public void set(String x, String y) {
        this.word = x;
        this.meaning = y;
    }

    @Override
    public void print() {
        System.out.println("국어사전");
        System.out.println(this.word + ": " + this.meaning);
    }
}

class KorEng implements Dic {
    String word;
    String meaning;

    KorEng(String word, String meaning) {
        this.word = word;
        this.meaning = meaning;
    }
```

```
    @Override
    public String get() {
        return this.meaning;
    }

    @Override
    public void set(String x, String y) {
        this.word = x;
        this.meaning = y;
    }

    @Override
    public void print() {
        System.out.println("영어사전");
        System.out.println(this.word + ": " + this.meaning);
    }
}

public class Dictionary {
    public static void main(String[] args) {
        KorKor k = new KorKor("오른쪽", "북쪽을 바라보고 동쪽과 같은 쪽");

        k.print();
        k.set("왼쪽", "북쪽을 바라보고 서쪽과 같은 쪽");
        System.out.println("왼쪽: " + k.get());
        KorEng e = new KorEng("하나", "one");
        e.print();
        e.set("둘", "two");
        System.out.println("둘: " + e.get());
    }
}
```

```
package midterm_exam;

import java.util.Arrays;
import java.util.Vector;

public class raw2ppm {
    public static void main(String[] args) {
        int [] r = {255,0,0, 0,255,0, 0,0,255};
        int [] g = {0,0,255, 0,255,0, 255,0,0};
        int [] b = {0,255,0, 255,255,255, 0,255,0};

        Vector v = new Vector<>();

        int[] ar = new int[r.length * 3];
```

```
        for(int i=0; i < r.length; i++) {
            v.add(r[i]);
            v.add(g[i]);
            v.add(b[i]);
        }

        for(int i=0; i < v.size(); i++) {
            ar[i] = (int) v.get(i);
        }

        System.out.print("P3 3 3 255\n");
        Arrays.stream(ar).forEach(s -> {
            System.out.print(s + " ");
        });
    }
}
```

```
package midterm_exam;

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

public class SignIn {
    // 윈도우
    Frame frame = new Frame("Adapter 테스트");

    Label idLabel = new Label("회원ID");
    TextField idTextField = new TextField();
    Button yesButton = new Button("처리");

    Label nameLabel = new Label("회원 이름");
    TextField nameTextField = new TextField();
    Button deleteButton = new Button("삭제");

    Label passwordLabel = new Label("패스워드");
    TextField passwordTextField = new TextField();
    Button exitButton = new Button("종료");

    Label birthLabel = new Label("주민등록번호");
    TextField birthTextField = new TextField();

    TextField freeTextField = new TextField(10);
    TextArea outScrollPane = new TextArea(10, 30);

    Button calculateButton = new Button("빈도수계산");
    Button resetButton = new Button("Reset");
```

```
    SignIn() {
        frame.setSize(500, 380);

        GridLayout gridLayout = new GridLayout(4, 3);
        Panel signEditPanel = new Panel();
        signEditPanel.setLayout(gridLayout);
        idLabel.setAlignment(Label.CENTER);
        nameLabel.setAlignment(Label.CENTER);
        passwordLabel.setAlignment(Label.CENTER);
        birthLabel.setAlignment(Label.CENTER);

        signEditPanel.add(idLabel);
        signEditPanel.add(idTextField);
        signEditPanel.add(yesButton);
        signEditPanel.add(nameLabel);
        signEditPanel.add(nameTextField);
        signEditPanel.add(deleteButton);
        signEditPanel.add(passwordLabel);
        signEditPanel.add(passwordTextField);
        signEditPanel.add(exitButton);
        signEditPanel.add(birthLabel);
        signEditPanel.add(birthTextField);

        Label blankLabel = new Label();

        Panel editTextPanel = new Panel();
        editTextPanel.setLayout(new BoxLayout(editTextPanel,
BoxLayout.Y_AXIS));

        yesButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                freeTextField.setText("");
                outScrollPane.setText(freeTextField.getText());
            }
        });

        deleteButton.addActionListener(s -> {
            freeTextField.setText("");
            outScrollPane.setText("");
            idTextField.setText("");
            nameTextField.setText("");
            passwordTextField.setText("");
            birthTextField.setText("");
        });

        editTextPanel.add(freeTextField);
```

```
        idTextField.addKeyListener(new KeyListener() {
            @Override
            public void keyTyped(KeyEvent e) {

            }

            @Override
            public void keyPressed(KeyEvent e) {
                if (idTextField.getText().equals("숫자 또는 영문자만 허용됨!")) {
                    idTextField.setText("");
                }
            }

            @Override
            public void keyReleased(KeyEvent e) {

                int key = e.getKeyChar();

                if ((key < 95 || key > 122) && (key < 65 || key > 90) && (key <
48 || key > 57) &&
                        key != 8 && key != 10) {
                    idTextField.setText("숫자 또는 영문자만 허용됨!");
                }

            }
        });

        nameTextField.addKeyListener(new KeyListener() {
            @Override
            public void keyTyped(KeyEvent e) {

            }

            @Override
            public void keyPressed(KeyEvent e) {
                if (nameTextField.getText().equals("영문자와 공백만 허용됨!")) {
                    nameTextField.setText("");
                }
            }

            @Override
            public void keyReleased(KeyEvent e) {
                int key = e.getKeyChar();

                if ((key < 95 || key > 122) && (key < 65 || key > 90) && key !=
32 &&
                        key != 8 && key != 10) {
                    nameTextField.setText("영문자와 공백만 허용됨!");
                }
```

```
            }
        });

        passwordTextField.addKeyListener(new KeyListener() {
            @Override
            public void keyTyped(KeyEvent e) {

            }

            @Override
            public void keyPressed(KeyEvent e) {
                if (passwordTextField.getText().equals("Space, Enter, Back
space 는 허용되지 않음!")) {
                    passwordTextField.setText("");
                }
            }

            @Override
            public void keyReleased(KeyEvent e) {
                int key = e.getKeyChar();

                if ( key == 32 || key == 10 || key == 8) {
                    passwordTextField.setText("Space, Enter, Back space 는 허용되
지 않음!");
                }
            }
        });

        birthTextField.addKeyListener(new KeyListener() {
            @Override
            public void keyTyped(KeyEvent e) {

            }

            @Override
            public void keyPressed(KeyEvent e) {
                if (birthTextField.getText().equals("숫자나 -만 허용됨!")) {
                    birthTextField.setText("");
                }

            }

            @Override
            public void keyReleased(KeyEvent e) {
                int key = e.getKeyChar();

                if ((key < 48 || key > 57) && key != 45 && key != 8 && key !=
10) {
                    birthTextField.setText("숫자나 -만 허용됨!");
```

```
            }
        }
    });

    exitButton.addActionListener(s -> {
        System.exit(0);
    });

    editTextPanel.add(blankLabel);
    editTextPanel.add(outScrollPane);

    Panel elseButtons = new Panel();
    elseButtons.setLayout(new BorderLayout());
    elseButtons.add(calculateButton, BorderLayout.WEST);
    elseButtons.add(resetButton, BorderLayout.EAST);

    frame.setLayout(new BoxLayout(frame, BoxLayout.Y_AXIS));
    frame.add(signEditPanel, BorderLayout.NORTH);
    frame.add(editTextPanel, BorderLayout.CENTER);
    frame.add(elseButtons);

    frame.setVisible(true);
    frame.addWindowListener(new WindowListener() {
        @Override
        public void windowOpened(WindowEvent e) {

        }

        @Override
        public void windowClosing(WindowEvent e) {
            System.exit(0);
        }

        @Override
        public void windowClosed(WindowEvent e) {

        }

        @Override
        public void windowIconified(WindowEvent e) {

        }

        @Override
        public void windowDeiconified(WindowEvent e) {

        }

        @Override
```

```
            public void windowActivated(WindowEvent e) {

            }

            @Override
            public void windowDeactivated(WindowEvent e) {

            }
        });
    }

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

```
package week5;

import java.awt.*;
import java.awt.event.*;
import java.lang.reflect.Array;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;

public class quiz {

    public abstract static class KBKeyPad extends Frame implements
ActionListener, WindowListener {
        Panel pbtn,ptf;
        Button b1,b2,b3,b4,b5,b6,b7,b8,b9,b0,b_cfrm,b_clear,b_del;

        TextField tf;
        StringBuffer sb = new StringBuffer(20);
        private String String;

        public KBKeyPad() {
            Frame f = new Frame("KeyPad");
            Label l = new Label("암호: ");

            pbtn = new Panel();
            tf = new TextField(20);
            ptf = new Panel();
            b1 = new Button();
            b2 = new Button();
            b3 = new Button();
            b4 = new Button();
```

```
        b5 = new Button();
        b6 = new Button();
        b7 = new Button();
        b8 = new Button();
        b9 = new Button();
        b0 = new Button();
        b_cfrm = new Button();
        b_clear = new Button();
        b_del = new Button();


        b1.setLabel("1");
        b2.setLabel("2");
        b3.setLabel("3");
        b4.setLabel("4");
        b5.setLabel("5");
        b6.setLabel("6");
        b7.setLabel("7");
        b8.setLabel("8");
        b9.setLabel("9");
        b0.setLabel("0");
        b_del.setLabel("하나지움");
        b_clear.setLabel("전체지움");
        b_cfrm.setLabel("확인");

        pbtn.add(l);
        pbtn.add(tf);

        ptf.add(b1);
        ptf.add(b2);
        ptf.add(b3);
        ptf.add(b4);
        ptf.add(b5);
        ptf.add(b6);
        ptf.add(b7);
        ptf.add(b8);
        ptf.add(b9);
        ptf.add(b_del);
        ptf.add(b0);
        ptf.add(b_clear);

        b1.addActionListener(this);
        b2.addActionListener(this);
        b3.addActionListener(this);
        b4.addActionListener(this);
        b5.addActionListener(this);
        b6.addActionListener(this);
        b7.addActionListener(this);
        b8.addActionListener(this);
```

```
        b9.addActionListener(this);
        b_del.addActionListener(this);
        b0.addActionListener(this);
        b_clear.addActionListener(this);
        b_cfrm.addActionListener(this);

        f.add(pbtn,BorderLayout.NORTH);
        f.add(ptf, BorderLayout.CENTER);
        f.add(b_cfrm, BorderLayout.SOUTH);

        ptf.setLayout(new GridLayout(4,3));

        f.setVisible(true);
        f.setSize(300,300);

        f.addWindowListener(new WindowEventHandler());
    }


    public static void main(String[] args) {
        new KBKeyPad() {
            @Override
            public void windowOpened(WindowEvent e) {

            }

            @Override
            public void windowClosed(WindowEvent e) {

            }

            @Override
            public void windowIconified(WindowEvent e) {

            }

            @Override
            public void windowDeiconified(WindowEvent e) {

            }

            @Override
            public void windowActivated(WindowEvent e) {

            }

            @Override
            public void windowDeactivated(WindowEvent e) {
```

```
                }
            };
        }

        public void resetNumber() {
            Button[] button = new Button[] {b0, b1, b2, b3, b4, b5, b6, b7, b8,
b9};
            String[] num = new String[]{"0", "1", "2", "3", "4", "5", "6", "7",
"8", "9"};
            ArrayList<String> number = new ArrayList<>(Arrays.asList(num));

            Collections.shuffle(number);

            for (int i = 0; i < 10; i++) {
                button[i].setLabel(number.get(i));
            }
        }

        public void actionPerformed(ActionEvent ae) {
            String name;
            name = ae.getActionCommand();
            if(name.equals("하나지움")) {
                sb.delete(sb.length()-1,sb.length());
                tf.setText(sb.toString());
            }

            else if(name.equals("전체지움")) {
                sb.delete(0,sb.length());
                tf.setText("");
            }

            else if(name.equals("확인"))
            {
                String pawd = "1234";
                if(pawd.equals(sb.toString())) {
                    sb.append(": Correct!");
                    tf.setText(sb.toString());
                }
                else
                {
                    sb.append(": Incorrect!");
                    tf.setText(sb.toString());
                }
            }

            else {
                sb.append(name);
                tf.setText(sb.toString());
                resetNumber();
```

```
            }
        }

        public void windowClosing(WindowEvent e) {
            System.exit(0);
        }

        class WindowEventHandler extends WindowAdapter {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        }

    }


}
```