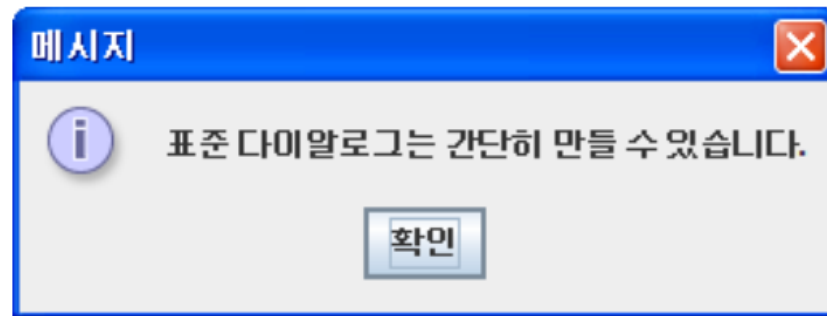


# 다이얼로그

# 대화 상자

- 대화 상자 윈도우는 임시 정보를 나타내는데 사용되는 독립적인 서브 윈도우이다.
- 몇 개의 미리 정해진 표준 대화 상자는 아주 간단히 만들 수 있다.

```
JOptionPane.showMessageDialog(frame, "표준 대화 상자는 간단히 만들 수 있습니다.");
```



# 팝업 다이얼로그, JOptionPane

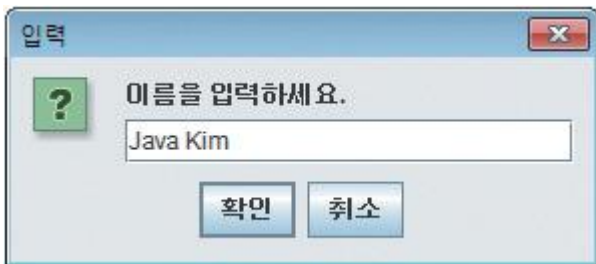
3

- 팝업 다이얼로그
  - ▣ 사용자에게 메시지를 전달하거나 문자열을 간단히 입력받는 용도
  - ▣ JOptionPane 클래스를 이용하여 생성
    - static 타입의 간단한 메소드 이용
- 입력 다이얼로그 - JOptionPane.showInputDialog()
- ▣ 한 줄을 입력 받는 다이얼로그

```
static String JOptionPane.showInputDialog(String msg)
```

- *msg* : 다이얼로그 메시지

- 리턴 값 : 사용자가 입력한 문자열. 취소 버튼이 선택되거나 창이 닫히면 null 리턴



```
String name = JOptionPane.showInputDialog("이름을 입력하세요.");  
// name에 "Java Kim"이 리턴  
// 취소 버튼이나, 입력 없이 다이얼로그가 닫히면 null 리턴
```

# 확인 다이얼로그

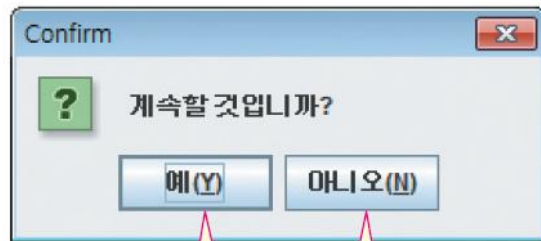
4

## □ 확인 다이얼로그 - JOptionPane.showConfirmDialog()

### ▣ 사용자로부터 Yes/No 응답을 입력 받는 다이얼로그

```
static int JOptionPane.showConfirmDialog(Component parentComponent, Object msg,  
                                         String title, int optionType)
```

- parentComponent : 다이얼로그의 부모 컴포넌트로서 다이얼로그가 출력되는 영역의 범위 지정을 위해 사용(예: 프레임). null이면 전체 화면 중앙에 출력
- msg : 다이얼로그 메시지
- title : 다이얼로그 타이틀
- optionType : 다이얼로그 옵션 종류 지정  
YES\_NO\_OPTION, YES\_NO\_CANCEL\_OPTION, OK\_CANCEL\_OPTION
- 리턴 값 : 사용자가 선택한 옵션 종류  
YES\_OPTION, NO\_OPTION, CANCEL\_OPTION, OK\_OPTION, CLOSED\_OPTION



```
int result = JOptionPane.showConfirmDialog(null, "계속할 것입니까?",  
                                           "Confirm", JOptionPane.YES_NO_OPTION);  
if(result == JOptionPane.CLOSED_OPTION) {  
    // 사용자가 "예", "아니오"의 선택 없이 다이얼로그 창을 닫은 경우  
}  
else if(result == JOptionPane.YES_OPTION) {  
    // 사용자가 "예"를 선택한 경우  
}  
else {  
    // 사용자가 "아니오"를 선택한 경우  
}
```

# 메시지 다이얼로그

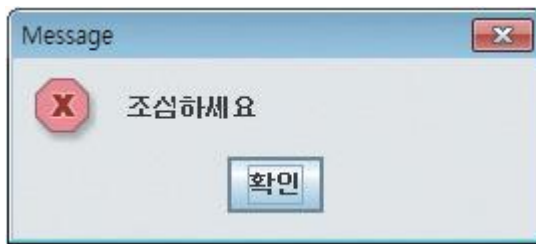
5

## □ 메시지 다이얼로그 – showMessageDialog

### ▣ 단순 메시지를 출력하는 다이얼로그

```
static void JOptionPane.showMessageDialog(Component parentComponent,  
                                           Object msg, String title, int messageType)
```

- parentComponent : 다이얼로그의 부모 컴포넌트로서 다이얼로그가 출력되는 영역의 범위 지정에 위해 사용(예: 프레임). null이면 전체 화면 중앙에 출력
- msg : 다이얼로그 메시지
- title : 다이얼로그 타이틀
- messageType : 다이얼로그의 종류로서 다음 중 하나  
ERROR\_MESSAGE, INFORMATION\_MESSAGE, WARNING\_MESSAGE, QUESTION\_MESSAGE,  
PLAIN\_MESSAGE



```
JOptionPane.showMessageDialog(null,  
    "조심하세요", "Message",  
    JOptionPane.ERROR_MESSAGE);
```

# 예제 : JOptionPane으로 3가지 팝업 다이얼로그 만들기

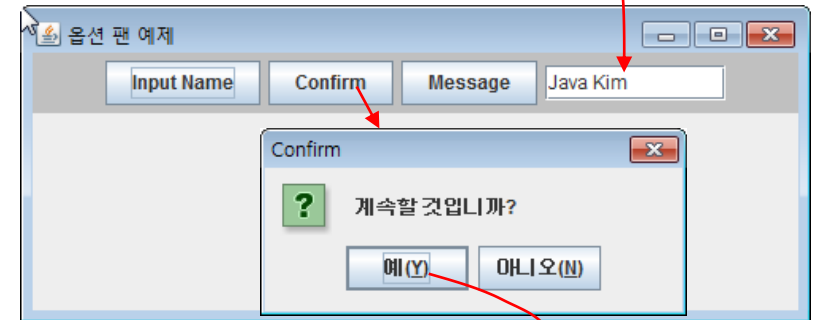
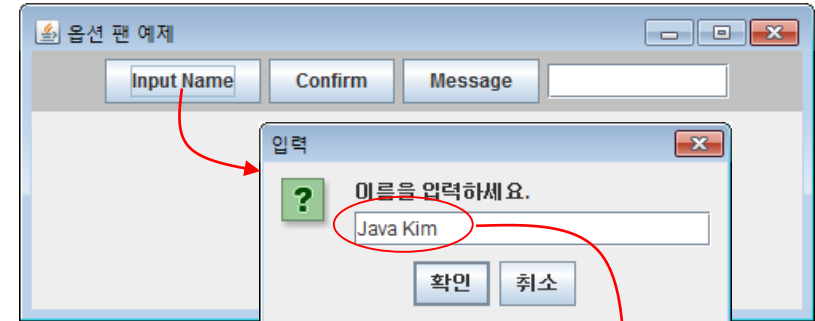
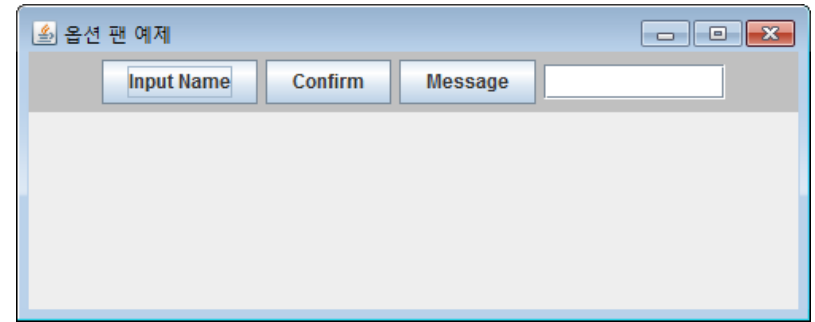
초기 화면

다음 그림과 같이  
3개의 팝업 다이얼로그를  
출력하는 응용프로그램을  
작성해보라

Input Name 버튼을 누르면  
입력 다이얼로그 생성.  
"Java Kim"을 입력하고  
확인 버튼을 누르면  
텍스트필드 창에 출력

Confirm 버튼을 누르면  
확인 다이얼로그 생성  
"예" 버튼을 누르면  
텍스트필드 창에 "Yes" 출력

Message 버튼을 누르면  
메시지 다이얼로그 생성  
"확인" 버튼을 누르면  
다이얼로그 종료



# 예제

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

public class OptionPaneEx extends JFrame {
    Container contentPane;

    OptionPaneEx() {
        setTitle("옵션 팬 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        contentPane = getContentPane();
        setSize(500,200);
        contentPane.add(new MyPanel(),
                        BorderLayout.NORTH);
        setVisible(true);
    }

    class MyPanel extends Panel {
        JButton inputBtn = new JButton("Input Name");
        JTextField tf = new JTextField(10);
        JButton confirmBtn = new JButton("Confirm");
        JButton messageBtn = new JButton("Message");

        MyPanel() {
            setBackground(Color.LIGHT_GRAY);
            add(inputBtn);
            add(confirmBtn);
            add(messageBtn);
            add(tf);
        }
    }
}
```

```
inputBtn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String name =
            JOptionPane.showInputDialog("이름을 입력하세요.");
        if(name != null)
            tf.setText(name);
    }
});

confirmBtn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        int result = JOptionPane.showConfirmDialog(null,
            "계속할 것입니까?", "Confirm",
            JOptionPane.YES_NO_OPTION);
        if(result == JOptionPane.CLOSED_OPTION)
            tf.setText("Just Closed without Selection");
        else if(result == JOptionPane.YES_OPTION)
            tf.setText("Yes");
        else
            tf.setText("No");
    }
});

messageBtn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null,
            "조심하세요", "Message",
            JOptionPane.ERROR_MESSAGE);
    }
});

}

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

# 대화 상자

## □ Dialog 클래스

- 메인 윈도우 외에 메시지를 출력하거나, 사용자로부터 데이터를 입력 받을 때 주로 사용하는 컨테이너이다.
- 보통은 Dialog 클래스로부터 상속을 받아 새로운 기능을 가진 대화상자를 만드는데 사용된다.

## □ Dialog 클래스의 생성자

생성자	설명
Dialog(Dialog owner)	생성되는 Dialog 객체를 소유하는 객체가 owner인 Frame을 생성한다.
Dialog(Dialog owner, String title)	생성되는 Dialog 객체의 소유자를 owner라는 객체로 설정하고 타이틀을 설정한다.
Dialog(Dialog owner, String title, boolean modal)	소유자로 owner 객체를 설정하고 타이틀을 가지며, 모달일지 모달이 아닌지를 설정하여 Dialog 객체를 생성한다.
Dialog(Frame owner)	생성되는 Dialog 객체를 소유하는 객체가 owner라는 Frame 객체를 생성한다.