



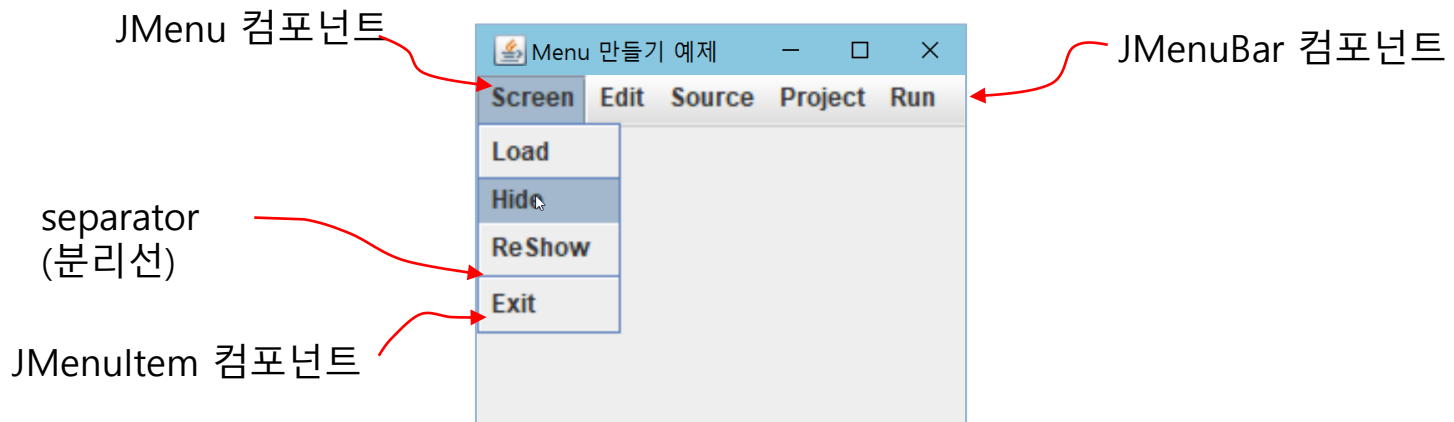
# 14

고급 스윙 컴포넌트

# 메뉴 만들기

2

- 메뉴 만들기에 필요한 스윙 컴포넌트
  - JMenuBar
    - 메뉴바의 기능을 하는 컴포넌트
  - JMenu
    - 파일, 편집 등 하나의 메뉴 기능을 하는 컴포넌트
  - JMenuItem
    - 파일 메뉴 내에 저장 등 메뉴 아이템 컴포넌트
  - 분리선
    - 메뉴 아이템 사이의 분리선



# 메뉴 만드는 과정

3

JMenuBar 컴포넌트



JMenuBar 컴포넌트



add



JMenu 컴포넌트

JMenuBar 컴포넌트



add



JMenu 컴포넌트



JMenuItem 컴포넌트

JMenuBar 컴포넌트



JMenu 컴포넌트



JMenu 컴포넌트

(1) JMenuBar 컴포넌트 생성

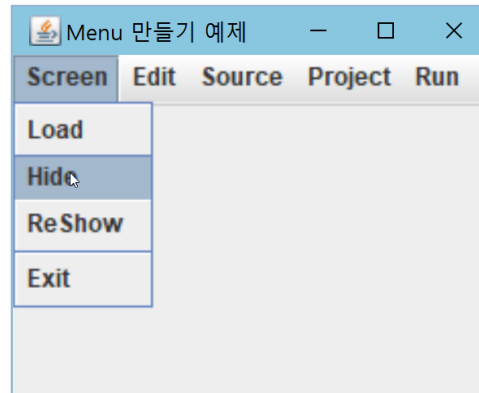
(2) JMenu 컴포넌트를  
생성하여 JMenuBar에  
붙인다.

(3) JMenuItem 컴포넌트를  
생성하여 JMenu에  
붙인다.

(3') 여러 개의 메뉴와  
메뉴 아이템을 생성한다.



(4) JMenuBar 컴포넌트를  
JFrame에 붙인다.



```
(1) → JMenuBar mb = new JMenuBar();

JMenu sMenu = new JMenu("Screen");
(2) → mb.add(sMenu);

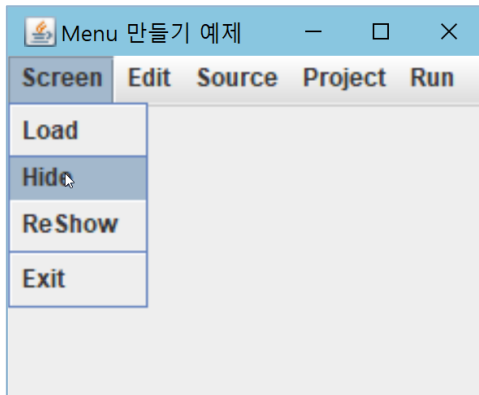
sMenu.add(new JMenuItem("Load"));
sMenu.add(new JMenuItem("Hide"));
sMenu.add(new JMenuItem("Re Show"));
sMenu.addSeparator();
sMenu.add(new JMenuItem("SaveAs"));
(3) →

(4) → frame.setJMenuBar(mb);
```

# 예제 14-1 : 메뉴 만들기

4

그림과 같이 Screen, Edit, Source, Project, Run의 5개 메뉴를 가지며, Screen 메뉴에만 4개의 메뉴아이템과 하나의 분리선(separator)을 가지도록 프로그램을 작성하라.



```
import javax.swing.*;
```

```
public class MenuEx extends JFrame {  
    public MenuEx() {  
        setTitle("Menu 만들기 예제");  
        createMenu(); // 메뉴 생성, 프레임에 삽입  
        setSize(250,200);  
        setVisible(true);  
    }  
    private void createMenu() {  
        JMenuBar mb = new JMenuBar();  
        JMenu screenMenu = new JMenu("Screen");  
  
        screenMenu.add(new JMenuItem("Load"));  
        screenMenu.add(new JMenuItem("Hide"));  
        screenMenu.add(new JMenuItem("ReShow"));  
        screenMenu.addSeparator(); // 분리선 삽입  
        screenMenu.add(new JMenuItem("Exit"));  
  
        mb.add(screenMenu);  
        mb.add(new JMenu("Edit"));  
        mb.add(new JMenu("Source"));  
        mb.add(new JMenu("Project"));  
        mb.add(new JMenu("Run"));  
  
        setJMenuBar(mb);  
    }  
    public static void main(String [] args) {  
        new MenuEx();  
    }  
}
```

# 메뉴아이템에 Action 이벤트 달기

5

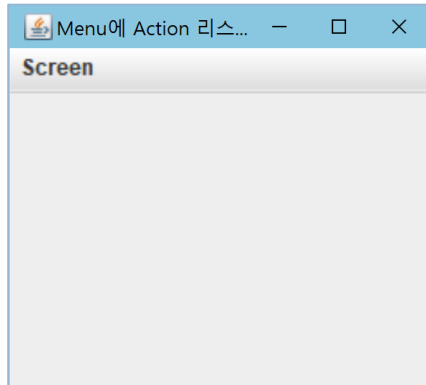
- ActionListener 이용
  - ▣ 각 JMenuItem 컴포넌트에 Action 이벤트 리스너 설정
    - 메뉴 아이템이 선택되었을 때 Action 이벤트 리스너 호출
- 메뉴 아이템에 ActionListener를 다는 예

```
JMenuItem item = new JMenuItem("Load");  
item.addActionListener(new MenuActionListener()); // 메뉴아이템에 Action 리스너 등록  
screenMenu.add(item);  
...  
  
class MenuActionListener implements ActionListener {  
    public void actionPerformed(ActionEvent e) {  
        // 사용자가 Load 메뉴아이템을 선택하는 경우 처리할 작업 구현  
    }  
}
```

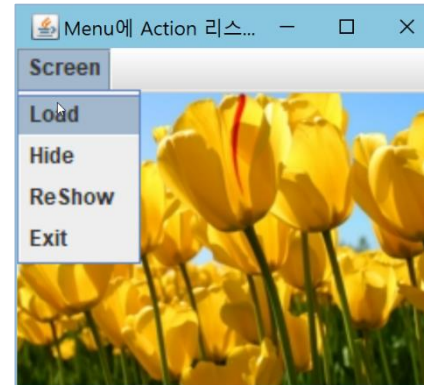


## 예제 14-2 : 메뉴아이템 선택 시 Action 리스너 활용

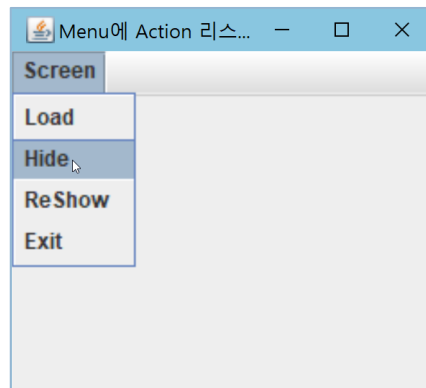
6



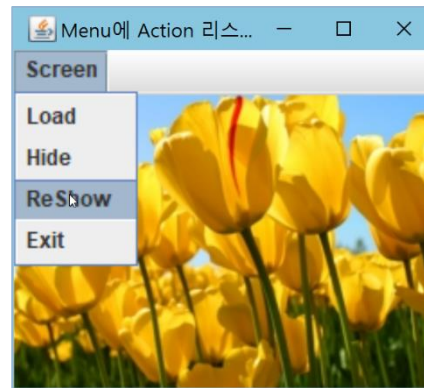
초기상태



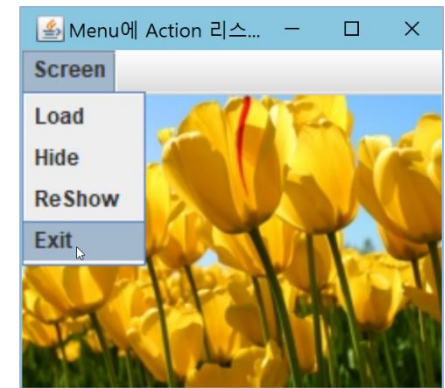
Load 메뉴아이템 선택하면  
레이블을 이용하여 이미지 출력



Hide 메뉴아이템 선택으로  
이미지를 보이지 않게 함



ReShow 메뉴아이템 선택으로  
숨겨진 이미지가 다시 보이게 함



Exit 메뉴아이템 선택하면  
프로그램 종료

## 예제 14-2 정답 코드

7

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

public class MenuActionEventEx extends JFrame {
    private JLabel imgLabel = new JLabel();
    public MenuActionEventEx() {
        setTitle("Menu에 Action 리스너 만들기 예제");
        createMenu();
        getContentPane().add(imgLabel, BorderLayout.CENTER);
        setSize(250,220);
        setVisible(true);
    }
    private void createMenu() {
        JMenuBar mb = new JMenuBar();
        JMenuItem [] menuItem = new JMenuItem [4];
        String[] itemTitle = {"Load", "Hide", "ReShow", "Exit"};
        JMenu screenMenu = new JMenu("Screen");

        // 4개의 메뉴아이템을 Screen 메뉴에 삽입한다.
        MenuActionListener listener = new MenuActionListener()
        for(int i=0; i<menuItem.length; i++) {
            menuItem[i] = new JMenuItem(itemTitle[i]);
            menuItem[i].addActionListener(listener);
            screenMenu.add(menuItem[i]);
        }
        mb.add(screenMenu);
        setJMenuBar(mb);
    }
}
```

```
class MenuActionListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        String cmd = e.getActionCommand();
        switch(cmd) { // 메뉴 아이템의 종류 구분
            case "Load" :
                if(imgLabel.getIcon() != null)
                    return; // 이미 로딩되었으면 리턴
                imgLabel.setIcon(new ImageIcon("images/img.jpg"));
                break;
            case "Hide" :
                imgLabel.setVisible(false);
                break;
            case "ReShow" :
                imgLabel.setVisible(true);
                break;
            case "Exit" :
                System.exit(0);
                break;
        }
    }
}

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

# 툴바

8

## □ JToolBar

### ▣ 툼바를 구현한 컴포넌트

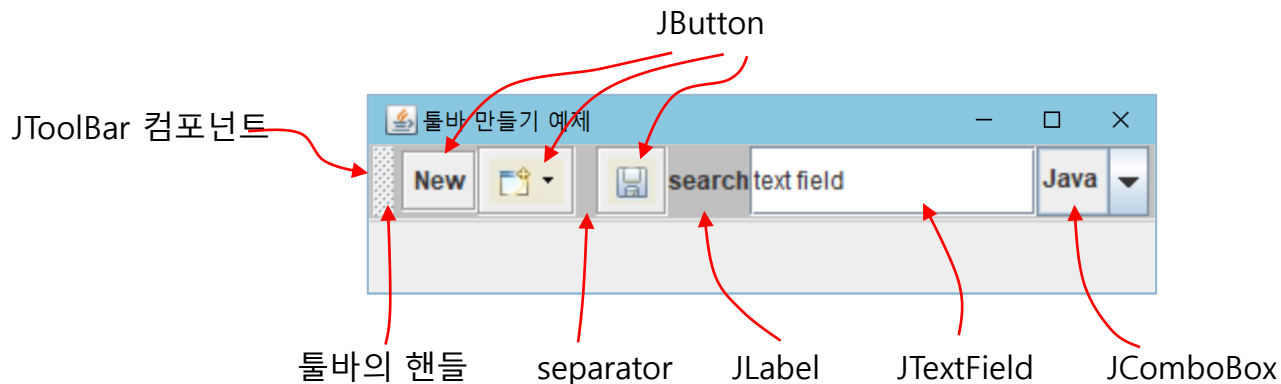
- 여러 컴포넌트를 담을 수 있는 컨테이너
- 버튼이나 이미지 등 모든 컴포넌트를 부착하여 이들을 메뉴처럼 사용

### ▣ 툼바의 모양

- 한 행 혹은 한 열로만 표현

### ▣ 툼바가 부착되는 위치

- 툼바는 BorderLayout 배치 관리자를 가진 컨테이너에만 부착
  - 상(NORTH), 하(SOUTH), 좌(WEST), 우(EAST) 측의 모서리 중 선택 부착
  - 사용자의 드래그에 의해 위의 4 가지 위치에 이동 부착 가능
- 사용자의 드래그에 의해 독립적인 다이얼로그 형태로 떨어져서 존재할 수 있음
- 사용자의 드래그에 의한 이동이 불가능하게 할 수 있음





# 툴바 만들기과 제어

9

## □ 툴바 만드는 과정

1. JToolBar 객체 생성

```
JToolBar toolBar = new JToolBar("Kitae Menu");
```

2. 컴포넌트를 JToolBar에 삽입

```
toolBar.add(new JButton("New")); // 버튼 삽입  
toolBar.addSeparator(); // 분리 공간 삽입  
toolBar.add(new JTextField("text field")); // 텍스트필드 삽입
```

3. JToolBar 컴포넌트를 컨테이너에 배치

- 컨테이너는 BorderLayout 배치 관리자

```
// 컨테이너의 NORTH에 툴바를 삽입한다.  
container.add(toolBar, BorderLayout.NORTH);
```

## □ 툴바 제어

### ■ 툴바 생성자

- JToolBar()
- JToolBar(String title)
  - 툴바가 윈도우에서 떨어져 나와 독립적으로 존재할 때 툴바의 타이틀로 *title* 문자열 설정

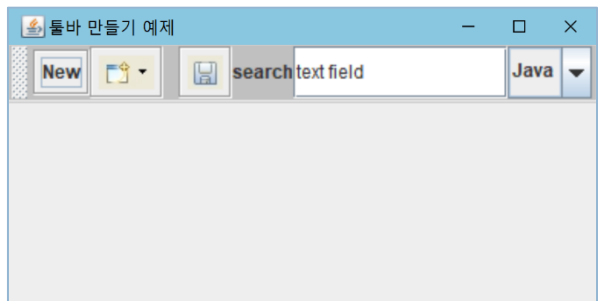
### ■ 툴바 드래그 못하게 고정하는 방법

- JToolBar.setFloatable(boolean b)
  - *b*가 *true*이면 드래그하여 이동 가능한 툴바, *false*이면 툴바의 핸들이 없어지며 드래그 불가능(이동 불가능)

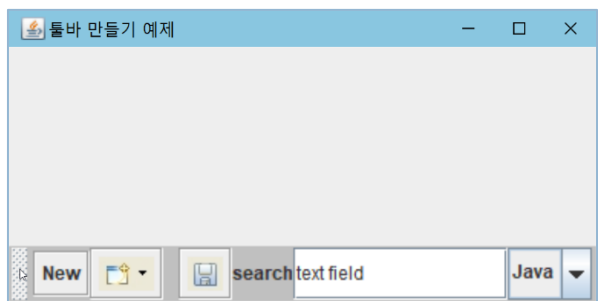
### ■ 툴바에 separator 삽입

- JToolBar.addSeparator()

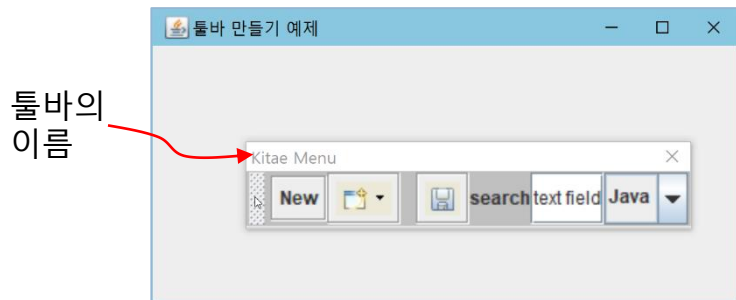
# 예제 14-3 : 툴바 만들기



초기 콘텐츠팬의 NORTH에 부착된 툴바



핸들을 드래그하여 SOUTH로 이동



핸들을 드래그하여 임의의 위치로 이동

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

public class ToolBarEx extends JFrame {
    private Container contentPane;
    public ToolBarEx() {
        setTitle("툴바 만들기 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        contentPane = getContentPane();
        createToolBar();
        setSize(400,200);
        setVisible(true);
    }

    private void createToolBar() {
        JToolBar toolBar = new JToolBar("Kitae Menu");
        bar.setBackground(Color.LIGHT_GRAY);

        toolBar.add(new JButton("New"));
        toolBar.add(new JButton(new ImageIcon("images/open.jpg")));
        toolBar.addSeparator();
        toolBar.add(new JButton(new ImageIcon("images/save.jpg")));
        toolBar.add(new JLabel("search"));
        toolBar.add(new JTextField("text field"));

        JComboBox combo = new JComboBox();
        combo.addItem("Java");
        combo.addItem("C#");
        combo.addItem("C");
        combo.addItem("C++");
        toolBar.add(combo);
        contentPane.add(toolBar, BorderLayout.NORTH);
    }

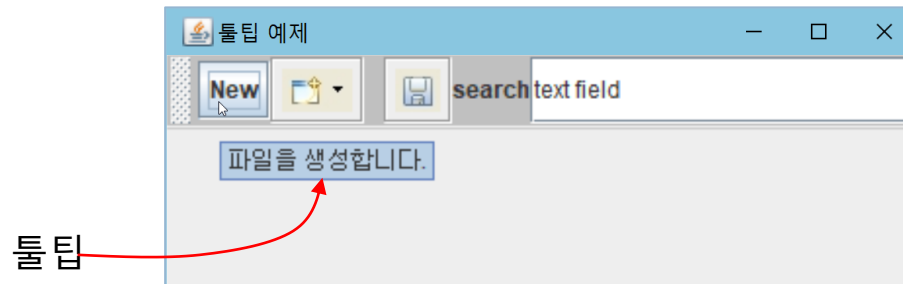
    public static void main(String [] args) {
        new ToolBarEx();
    }
}
```

# 툴 팁

11

## □ 툴 팁

- 스윙 컴포넌트를 설명하는 팁 문구
  - 팁 문구는 문자열만 가능
- 컴포넌트 위에 마우스를 올리면 잠깐 나타났다가 사라짐

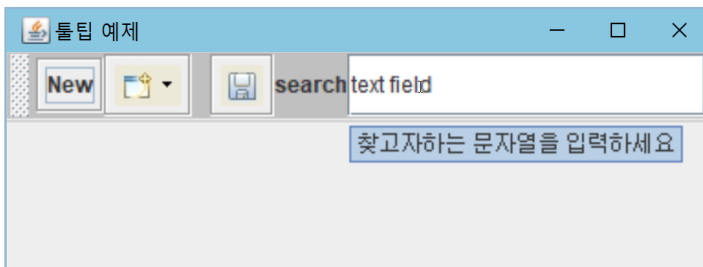
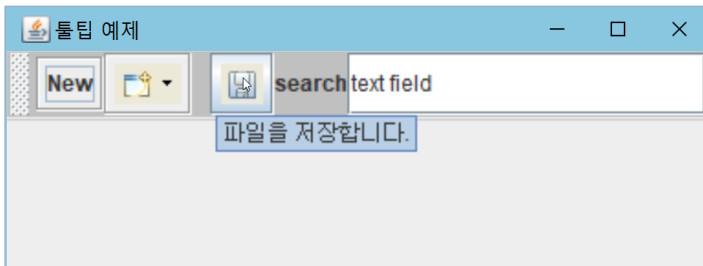
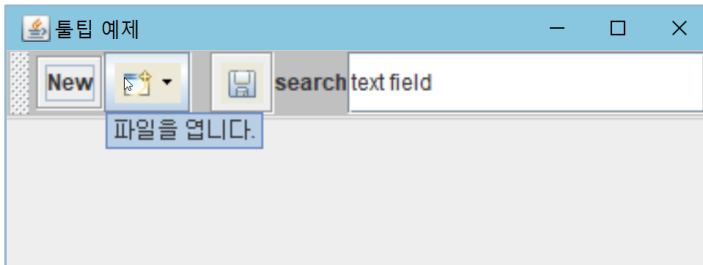
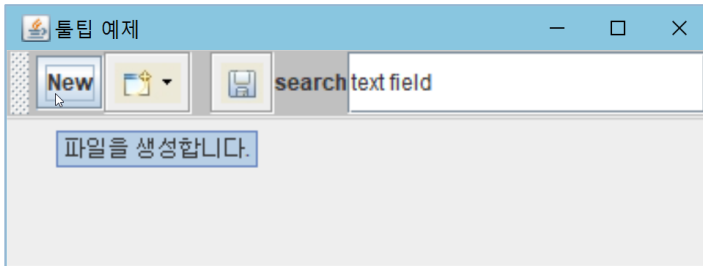


## □ 툴 팁 생성

1. ~~JToolTip 클래스 이용(복잡하므로 권하지 않음)~~
2. JComponent.setToolTipText(String msg) 이용 : 권장

```
// 버튼에 툴팁을 다는 예
JButton b = new JButton("open");
b.setToolTipText("클릭하면 파일을 엽니다");
```

# 예제 14-4 : 툴팁 달기



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

```
public class ToolTipEx extends JFrame {
    private Container contentPane;
    public ToolTipEx() {
        setTitle("툴팁 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        contentPane = getContentPane();
        createToolBar();
        setSize(400,150);
        setVisible(true);
    }
}
```

```
private void createToolBar() {
    JToolBar bar = new JToolBar("Kitae Menu");
    bar.setBackground(Color.LIGHT_GRAY);
    JButton newBtn = new JButton("New");
    newBtn.setToolTipText("파일을 생성합니다.");
    bar.add(newBtn);
}
```

```
JButton openBtn = new JButton(new ImageIcon("images/open.jpg"));
openBtn.setToolTipText("파일을 엽니다.");
bar.add(openBtn);
bar.addSeparator();
}
```

```
JButton saveBtn = new JButton(new ImageIcon("images/save.jpg"));
saveBtn.setToolTipText("파일을 저장합니다.");
bar.add(saveBtn);
}
```

```
bar.add(new JLabel("search"));
```

```
JTextField tf = new JTextField("text field");
tf.setToolTipText("찾고자하는 문자열을 입력하세요");
bar.add(tf);
contentPane.add(bar, BorderLayout.NORTH);
}
```

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

# 툴팁 활성화 및 툴팁 지연 시간 제어

13

- ▣ ToolTipManager 클래스 이용
  - 툴팁 제어 기능은 모든 툴팁에 일괄 적용
- ▣ 툴팁 매니저 객체 생성

```
ToolTipManager m = ToolTipManager.sharedInstance();
```

- ▣ 툴팁 활성화/비활성화

```
m.setEnabled(false); // 툴팁이 보이지 않게 함
```

- ▣ 툴팁이 나타나는 최초 지연시간 제어

```
m.setInitialDelay(1000) // 마우스가 올라온 후 1000 ms 후에 툴팁 출력
```

- ▣ 툴팁이 켜져 있는 지연 시간 제어

```
m.setDismissDelay(1000) // 툴팁이 켜져 있는 지속 시간을 1000ms로 설정
```

# 예제 14-5 : 툴팁 시간 제어

JLabel 컴포넌트에 마우스를 올리자  
마자 툴팁이 나타나게 하고, 툴팁은  
10초간 지속되게 하라.



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

public class ToolTipDelayEx extends JFrame {
    public ToolTipDelayEx() {
        setTitle("툴팁 지연 시간 제어 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        c.setLayout(new FlowLayout());
        JLabel cherryLabel = new JLabel(new
                                     ImageIcon("images/cherry.jpg"));
        cherryLabel.setToolTipText("체리 이미지 어때요");
        JLabel appleLabel = new JLabel(new ImageIcon("images/apple.jpg"));
        appleLabel.setToolTipText("사과 이미지 어때요");
        c.add(cherryLabel);
        c.add(appleLabel);

        ToolTipManager m = ToolTipManager.sharedInstance();
m.setInitialDelay(0);
m.setDismissDelay(10000);

        setSize(400,200);
        setVisible(true);
    }

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

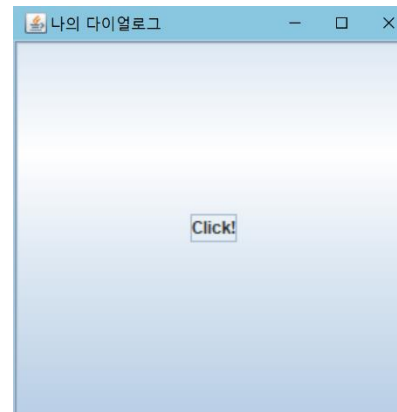


# 다이얼로그 만들기

15

- 다이얼로그란?
  - ▣ 사용자로부터 입력을 받기 위한 대화 상자
- JDialog
  - ▣ 스윙에서 다이얼로그를 만들기 위한 슈퍼 클래스
  - ▣ 최상위 컨테이너(top-level container)
    - 다른 컨테이너에 속할 필요 없이 화면에 출력 가능한 컨테이너
- 300x300 크기의 다이얼로그를 만드는 간단한 코드

```
JDialog dialog = new JDialog();  
dialog.setTitle("나의 다이얼로그");  
dialog.add(new JButton("click!"));  
dialog.setSize(300,300);  
dialog.setVisible(true);
```



# 예제 14-6 : JDialog를 상속받아 다이얼로그 만들기

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

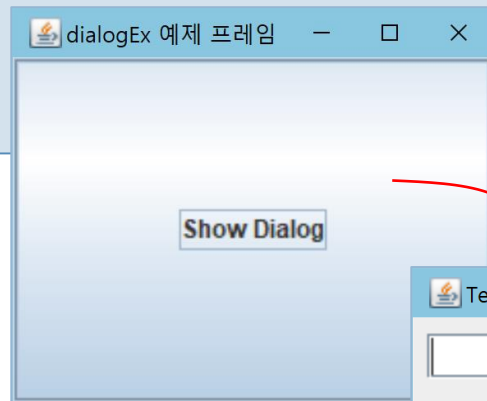
class MyDialog extends JDialog {
    private JTextField tf = new JTextField(10);
    private JButton okButton = new JButton("OK");

    public MyDialog(JFrame frame, String title) {
        super(frame, title);
        setLayout(new FlowLayout());
        add(tf);
        add(okButton);
        setSize(200, 100);

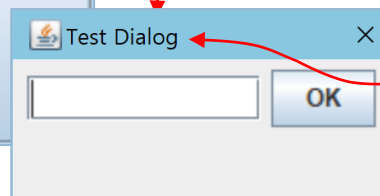
        okButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                setVisible(false);
            }
        });
    }
}
```

```
public class DialogEx extends JFrame{
    private MyDialog dialog;
    public DialogEx() {
        super("DialogEx 예제 프레임");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JButton btn = new JButton("Show Dialog");

        dialog = new MyDialog(this, "Test Dialog");
        btn.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                dialog.setVisible(true);
            }
        });
        getContentPane().add(btn);
        setSize(250,200);
        setVisible(true);
    }
    public static void main(String[] args) {
        new DialogEx();
    }
}
```



버튼을 선택하면  
"Test Dialog"  
다이얼로그가 생성된다.



다이얼로그의 타이틀

MyDialog로 만든 다이얼로그

# 모달 다이얼로그와 모달리스 다이얼로그

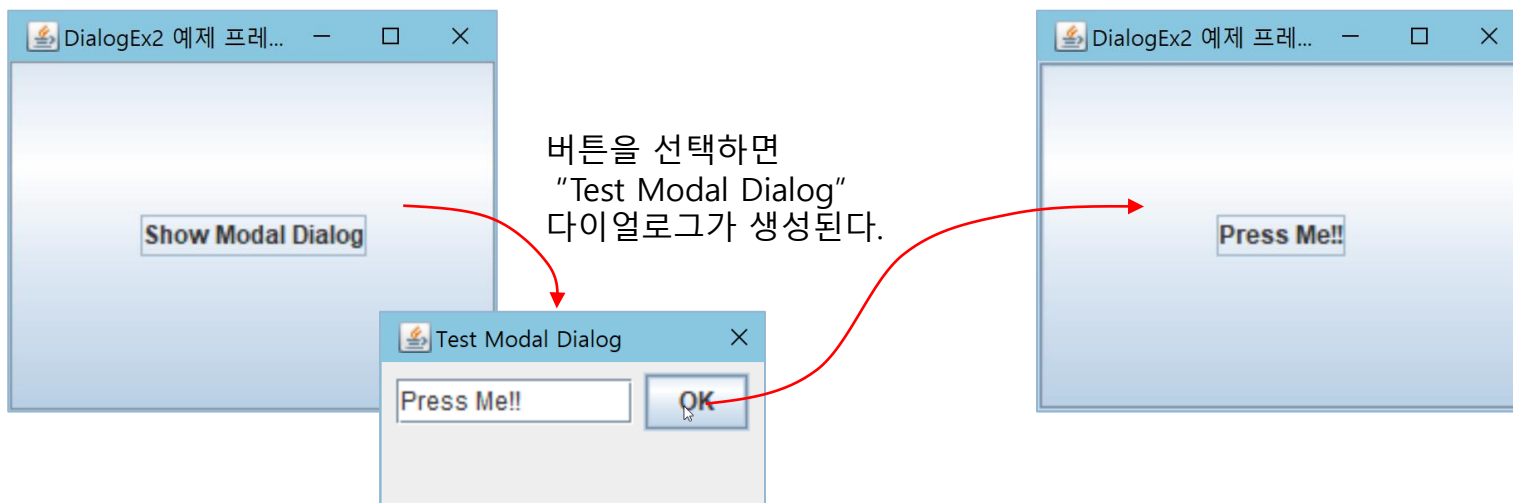
17

- 모달(modal) 다이얼로그
  - ▣ 사용자 입력을 독점하는 다이얼로그
  - ▣ 닫기 전에는 다른 창에서 작업을 전혀 할 수 없는 다이얼로그
  - ▣ 생성 방법
    - `JDialog(Frame owner, String title, boolean modal)` 생성자에서 `modal`을 `true` 값으로 지정
- 모달리스 다이얼로그
  - ▣ 다른 창과 독립적으로 작동하는 다이얼로그

# 예제 14-7: 모달 다이얼로그로부터 사용자의 입력 값 알아내기

18

텍스트필드 창과 OK 버튼을 가진 모달 다이얼로그를 만들고, 사용자가 텍스트필드 창에 문자열을 입력하고 OK 버튼을 클릭하면 입력한 문자열을 Show Modal Dialog 버튼의 문자열을 변경하도록 예제 14-6을 수정하라.



사용자가 "Press Me!!"라고 입력한 후  
OK 버튼을 누르면  
버튼의 문자열이 "Press Me!!"로 변경된다.

## 예제 14-7 : 정답 코드

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

class MyModalDialog extends JDialog {
    private JTextField tf = new JTextField(10);
    private JButton okButton = new JButton("OK");

    public MyModalDialog(JFrame frame, String title) {
        super(frame, title, true); // 모달 다이얼로그로 설정
        setLayout(new FlowLayout());
        add(tf);
        add(okButton);
        setSize(200, 100);

        okButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                setVisible(false);
            }
        });
    }

    public String getInput() {
        if(tf.getText().length() == 0) return null;
        else return tf.getText();
    }
}
```

```
public class DialogEx2 extends JFrame {
    private MyModalDialog dialog;
    public DialogEx2() {
        super("DialogEx2 예제 프레임");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JButton btn = new JButton("Show Modal Dialog");

        dialog = new MyModalDialog(this, "Test Modal Dialog");

        btn.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                dialog.setVisible(true);

                String text = dialog.getInput();

                if(text == null) return;
                JButton btn = (JButton)e.getSource();
                btn.setText(text);
            }
        });
        getContentPane().add(btn);
        setSize(250, 200);
        setVisible(true);
    }
    public static void main(String[] args) {
        new DialogEx2();
    }
}
```

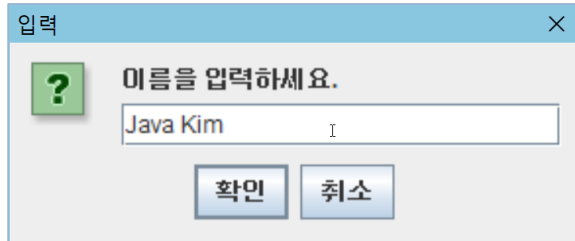
# 팝업 다이얼로그, JOptionPane

20

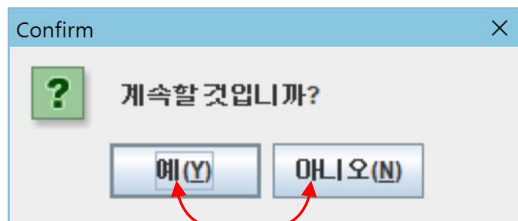
- JOptionPane
  - ▣ 간단하고 유용한 팝업 다이얼로그 지원
    - static 의 타입의 간단한 메소드 이용
- 입력 다이얼로그
  - ▣ 한 줄을 입력 받는 다이얼로그
  - ▣ `String JOptionPane.showInputDialog(String msg)`
- 확인 다이얼로그
  - ▣ 사용자로부터 Yes/No 등 확인을 입력받는 다이얼로그
  - ▣ `int JOptionPane.showConfirmDialog(Component parentComponent, Object msg, String title, int optionType)`
- 메시지 다이얼로그
  - ▣ 단순 메시지를 출력하는 다이얼로그
  - ▣ `void JOptionPane.showMessageDialog(Component parentComponent, Object msg, String title, int messageType)`



# JOptionPane 팝업 다이얼로그와 코드 샘플

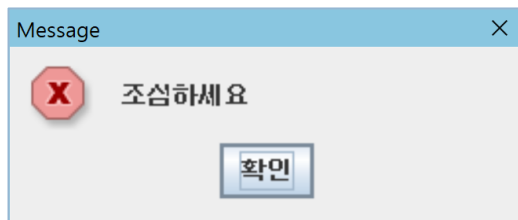


```
String name = JOptionPane.showInputDialog("이름을 입력하세요.");  
// name에 "Java Kim" 리턴  
// 사용자가 입력 없이 창을 닫으면 null 리턴
```



옵션  
(JOptionPane.YES\_NO\_OPTION)

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



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

# 예제 14-8 : JOptionPane을 사용한 팝업 다이얼로그 작성

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

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

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

        public 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);
    }
});

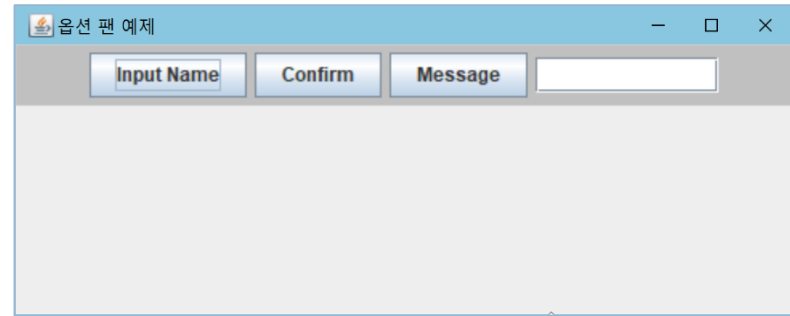
}

}

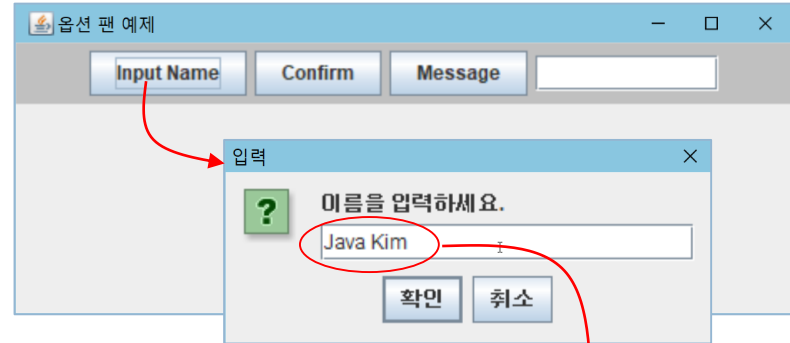
public static void main(String [] args) {
    new OptionPaneEx();
}
}
```

# 예제 실행

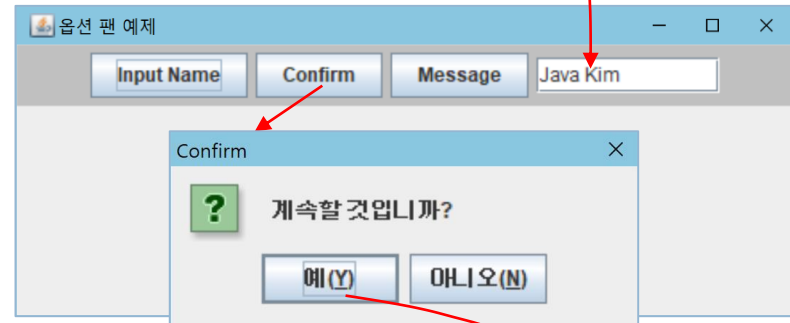
초기 화면



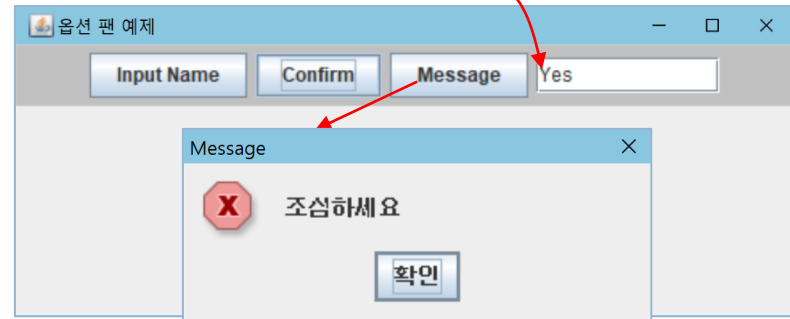
Input Name 버튼을 누르는 경우 입력 다이얼로그 생성  
"Java Kim"을 입력하고 확인 버튼을 누르면  
텍스트필드 창에 출력



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



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



# 파일 다이얼로그

24

## □ JFileChooser

- 파일 시스템의 탐색기와 같은 기능을 하는 다이얼로그
- 사용자에게 파일이나 디렉터리를 쉽게 선택하도록 하는 기능
- 다이얼로그 종류
  - 파일 열기 다이얼로그(File Open Dialog)
  - 파일 저장 다이얼로그(File Save Dialog)

## □ 주의 사항

- 파일 다이얼로그에서 사용자가 파일을 선택하는 행위란?
  - 사용자가 선택한 파일 이름을 얻기 위함
  - 파일을 읽거나 쓰는 행위가 아님
- 다이얼로그 닫기란?
  - 다이얼로그가 화면에서 보이지 않게 됨
  - JFileChooser 객체가 없어지거나 사용할 수 없게 것이 아님
    - `showOpenDialog()` 메소드를 호출하면 다시 화면에 출력하고 재사용 가능
  - 다이얼로그가 닫힌 후 JFileChooser 객체로부터 사용자가 입력한 정보를 알아낼 수 있음

# JFileChooser 파일 다이얼로그, 코드 샘플

## 파일 열기 다이얼로그 생성 과정

1. JFileChooser 객체 생성

```
JFileChooser chooser = new JFileChooser();
```

2. 파일 필터  
객체 생성

```
FileNameExtensionFilter filter = new FileNameExtensionFilter(  
    "JPG & GIF",  
    "jpg", "gif");
```

3. JFileChooser에  
파일 필터 설정

생략가능

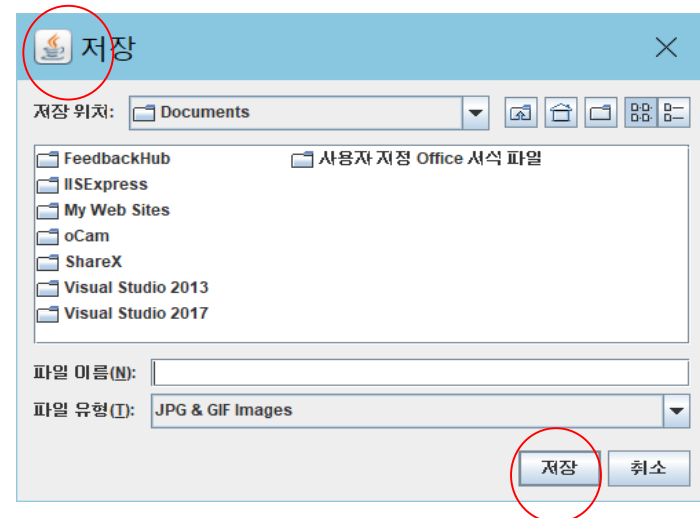
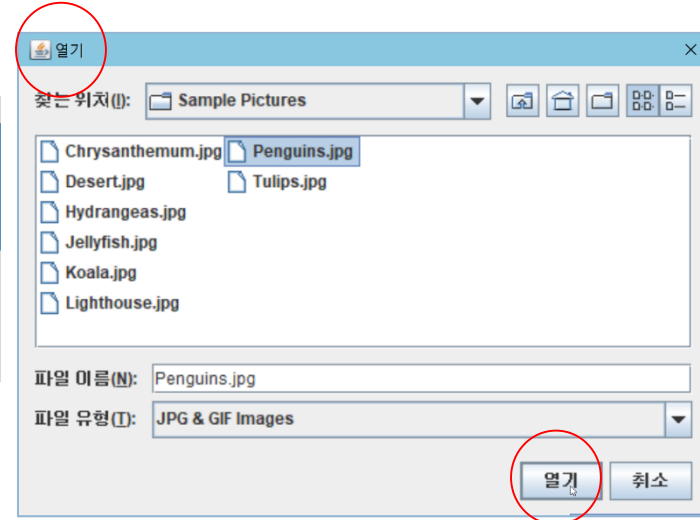
```
chooser.setFileFilter(filter);
```

4. 열기 다이얼로그 출력

```
int ret = chooser.showOpenDialog(null);
```

5. 사용자가 선택한  
파일 이름 알아내기

```
if(ret == JFileChooser.APPROVE_OPTION) {  
    String pathName = chooser.getSelectedFile().getPath();  
    String fileName = chooser.getSelectedFile().getName();  
}
```

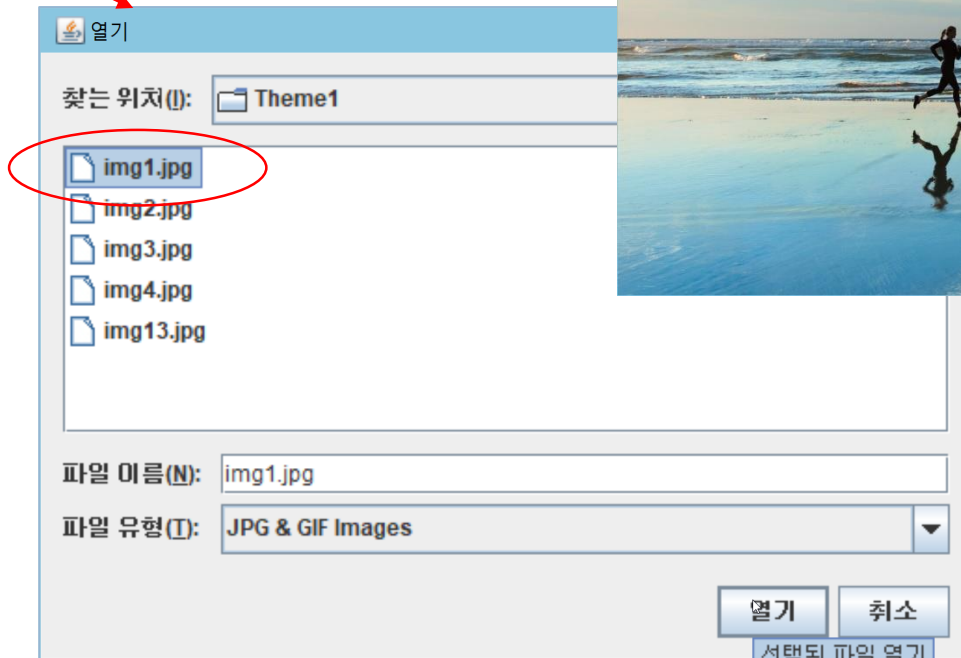
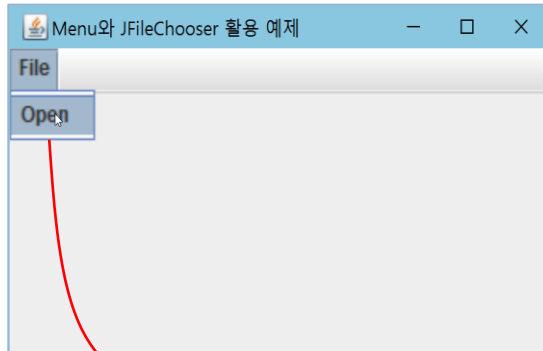


## 파일 저장 다이얼로그 생성 과정

```
int ret = chooser.showSaveDialog(null);
```

# 예제 14-9 : 파일 열기 다이얼로그 생성 및 출력

26





## 예제 14-9 : 정답 코드

27

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

public class MenuAndFileDialogEx extends JFrame {
    private JLabel imageLabel = new JLabel();
    public MenuAndFileDialogEx() {
        setTitle("Menu와 JFileChooser 활용 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        c.add(imageLabel);
        createMenu();
        setSize(350,200);
        setVisible(true);
    }

    private void createMenu() {
        JMenuBar mb = new JMenuBar();
        JMenu fileMenu = new JMenu("File");
        JMenuItem openItem = new JMenuItem("Open");

        openItem.addActionListener(
            new OpenActionListener());
        fileMenu.add(openItem);
        mb.add(fileMenu);
        setJMenuBar(mb);
    }
}
```

```
class OpenActionListener implements ActionListener {
    private JFileChooser chooser;

    public OpenActionListener() {
        chooser = new JFileChooser();
    }
    public void actionPerformed(ActionEvent e) {
        FileNameExtensionFilter filter =
            new FileNameExtensionFilter("JPG & GIF Images",
                "jpg", "gif");
        chooser.setFileFilter(filter);

        int ret = chooser.showOpenDialog(null);
        if(ret != JFileChooser.APPROVE_OPTION) {
            JOptionPane.showMessageDialog(null,
                "파일을 선택하지 않았습니다", "경고",
                JOptionPane.WARNING_MESSAGE);
            return;
        }

        String filePath = chooser.getSelectedFile().getPath();
        imageLabel.setIcon(new ImageIcon(filePath));
        pack(); // 이미지의 크기에 맞추어 프레임 크기 조절
    }
    public static void main(String [] args) {
        new MenuAndFileDialogEx();
    }
}
```

# 컬러 다이얼로그

28

## □ JColorChooser

- ▣ 색상 팔레트를 제공하는 모달 다이얼로그

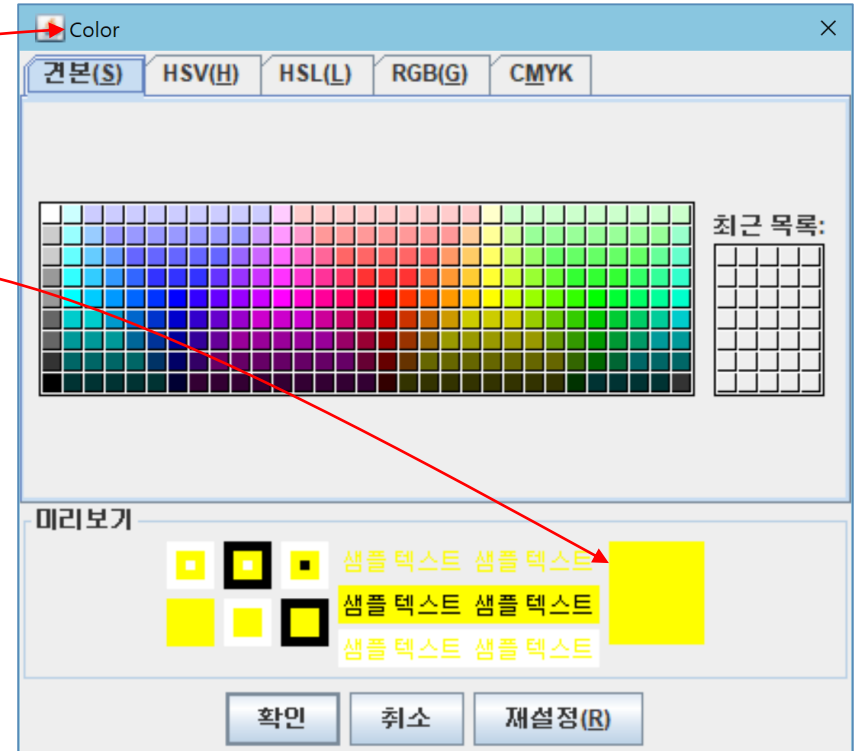
### 컬러 다이얼로그 출력 과정

#### 1. 컬러 다이얼로그 출력

```
Color selectedColor = JColorChooser.showDialog(  
    null,  
    "Color",  
    Color.YELLOW);  
// 사용자가 "확인" 버튼을 누르면 선택한 색을 리턴하지만  
// 사용자가 "취소" 버튼이나 다이얼로그를 강제로 닫을 때  
// null 리턴
```

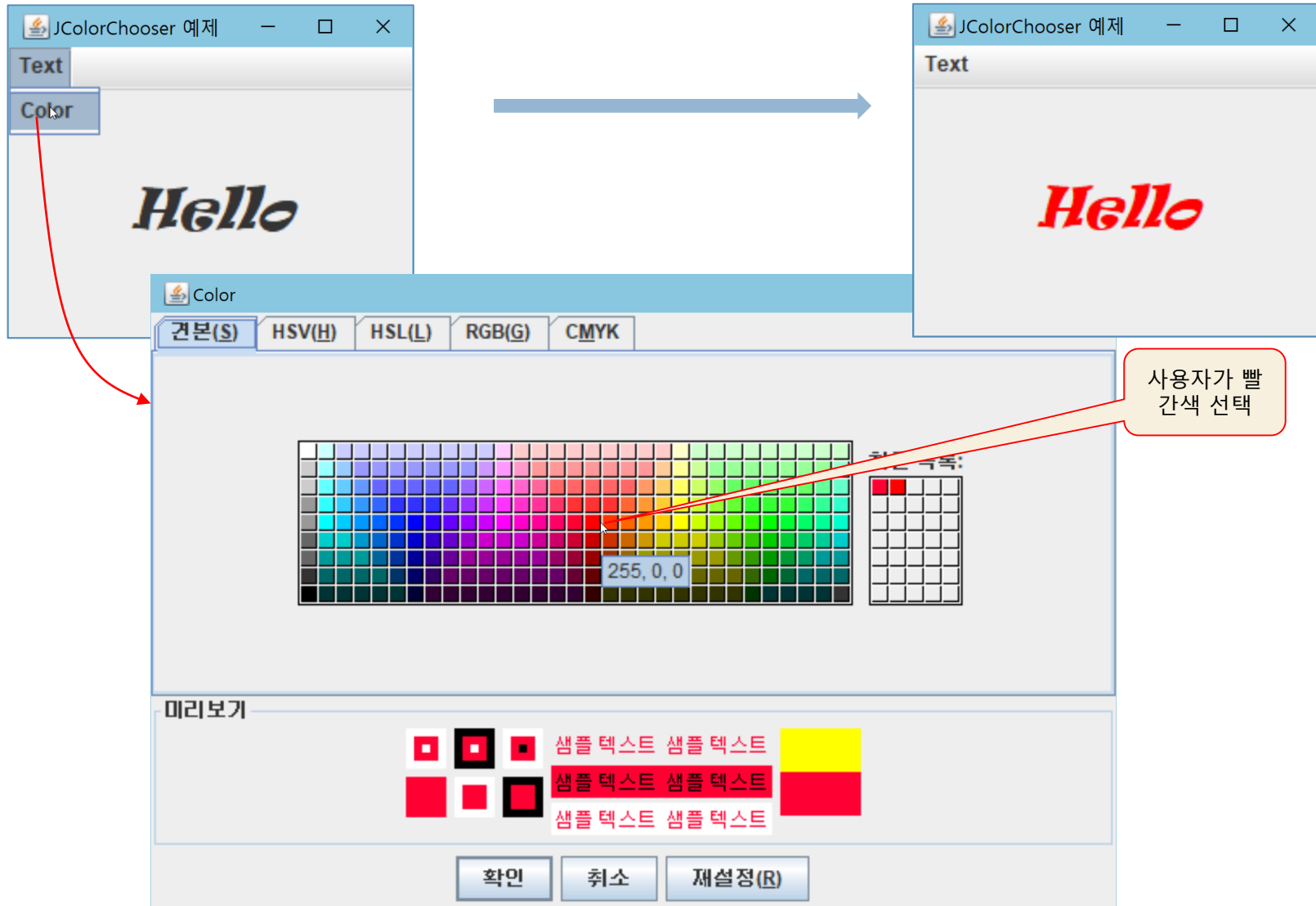
#### 3. 사용자가 선택한 색 알아내기

```
if(selectedColor != null) {  
    // 사용자가 선택한 색 selectedColor 사용  
}
```



# 예제 14-10 : JColorChooser를 이용한 컬러 다이얼로그 사용

29



# 예제 14-10 : 정답 코드

30

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

public class MenuAndColorChooserEx extends JFrame {
    private JLabel label = new JLabel("Hello");

    public MenuAndColorChooserEx() {
        setTitle("JColorChooser 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        label.setHorizontalAlignment(SwingConstants.CENTER);
        label.setFont(new Font("Ravie", Font.ITALIC, 30));
        c.add(label, BorderLayout.CENTER);
        createMenu();
        setSize(250,200);
        setVisible(true);
    }

    private void createMenu() {
        JMenuBar mb = new JMenuBar();
        JMenuItem colorMenuItem = new JMenuItem("Color");
        JMenu fileMenu = new JMenu("Text");
        colorMenuItem.addActionListener(new
            MenuActionListener());

        fileMenu.add(colorMenuItem);
        mb.add(fileMenu);
        this.setJMenuBar(mb);
    }
}
```

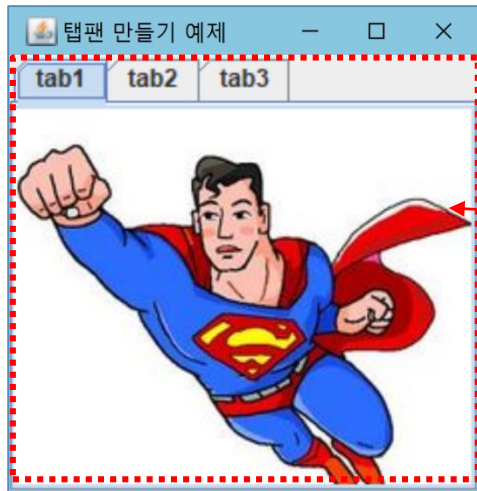
```
class MenuActionListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        String cmd = e.getActionCommand();
        if(cmd.equals("Color")) {
            Color selectedColor =
                JColorChooser.showDialog(null, "Color",
                    Color.YELLOW);

            if(selectedColor != null)
                label.setForeground(selectedColor);
        }
    }
}

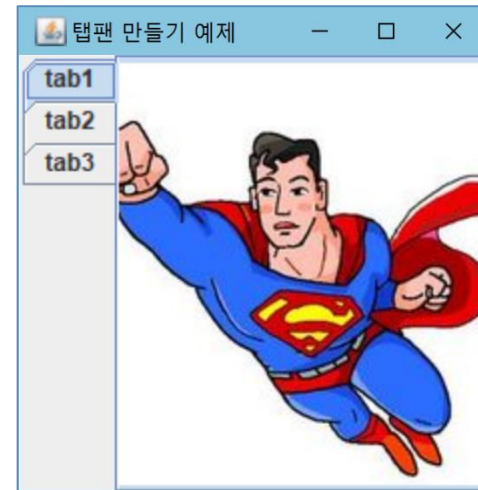
public static void main(String [] args) {
    new MenuAndColorChooserEx();
}
```

## □ JTabbedPane

- ▣ 여러 패널을 겹치게 하여 공간을 공유하도록 지원하는 팬



3 개의 탭을 가진  
JTabbedPane



탭 위치(tabPlacement)가  
JTabbedPane.LEFT인 경우

## ▣ 생성자

- JTabbedPane()
- JTabbedPane(int tabPlacement)
  - *tabPlacement* : JTabbedPane.TOP, JTabbedPane.BOTTOM, JTabbedPane.LEFT, JTabbedPane.RIGHT

# 탭팬 주요 메소드

32

## □ 탭 만들어 탭팬에 붙이기

- void addTab(String title, Component comp)
  - title : 탭의 이름, comp : 탭을 구성하는 컴포넌트

```
JTabbedPane pane = new JTabbedPane(JTabbedPane.LEFT);  
pane.addTab("tab1", new JLabel(new ImageIcon("images/img1.jpg")));  
pane.addTab("tab2", new JLabel(new ImageIcon("images/img2.jpg"));  
pane.addTab("tab3", new MyPanel();  
add(pane);
```

## □ 탭팬에 붙은 탭의 개수

- int getTabCount() - 탭팬에 붙은 탭의 개수 리턴

## □ 현재 선택된 탭

- int getSelectedIndex()
- Component getSelectedComponent()

## □ 탭 삭제

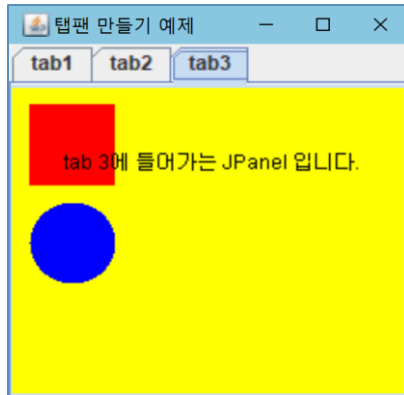
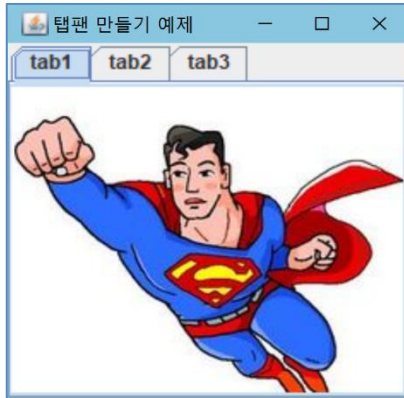
- void remove(Component comp)
- void remove(int index)
- void removeTabAt(int index)

## □ 탭 위치 변경

- void setTabPlacement(int tabPlacement)



# 예제 14-11 : 탭팬 만들기



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

```
public class TabbedPaneEx extends JFrame {
    public TabbedPaneEx() {
        setTitle("탭팬 만들기 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        JTabbedPane pane = createTabbedPane();
        c.add(pane, BorderLayout.CENTER);
        setSize(250,250);
        setVisible(true);
    }

    private JTabbedPane createTabbedPane() {
        JTabbedPane pane = new JTabbedPane(JTabbedPane.LEFT);
        pane.addTab("tab1", new JLabel(new ImageIcon("images/img1.jpg")));
        pane.addTab("tab2", new JLabel(new ImageIcon("images/img2.jpg")));
        pane.addTab("tab3", new MyPanel());
        return pane;
    }

    class MyPanel extends JPanel {
        public MyPanel() {
            this.setBackground(Color.YELLOW);
        }
        public void paintComponent(Graphics g) {
            super.paintComponent(g);
            g.setColor(Color.RED);
            g.fillRect(10,10,50,50);
            g.setColor(Color.BLUE);
            g.fillOval(10,70,50,50);
            g.setColor(Color.BLACK);
            g.drawString("tab 3에 들어가는 JPanel 입니다.", 30, 50);
        }
    }

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

# 자바의 오디오 API

34

- 자바의 오디오 API로 다룰 수 있는 오디오 종류
  - ▣ 디지털 오디오(Digital Audio)
    - 음악, 목소리 등의 아날로그 데이터를 디지털로 만든 오디오
    - WAV, AU, AIFF, AIFC
  - ▣ 미디(MIDI: Music Instrument Digital Interface) 데이터
    - 피아노, 바이올린 등 악기 소리와 높이 등을 표현하는 오디오
    - 미디 장치에서 연주

# 디지털 오디오 재생 과정

35

1. 오디오 클립 만들기 : Clip 인터페이스 이용

```
Clip clip = AudioSystem.getClip();
```

2. 오디오 클립에 오디오 스트림 연결.  
오디오 파일로부터 오디오를 읽을  
스트림 객체 생성

```
File audioFile = new File("애국가.wav");  
AudioInputStream audioStream =  
    AudioSystem.getAudioInputStream(audioFile);
```

오디오 클립에 오디오 스트림 연결

```
clip.open(audioStream);
```

3. 오디오 재생  
Clip 클래스의 start() 메소드 호출

```
clip.start(); // 오디오 재생 시작
```

```
void open(AudioInputStream stream)
```

오디오 클립은 stream에 존재하는 오디오의 형식과 데이터를 인식하고, 재생할 수 있는 준비를 갖춘다.

```
void start()
```

현재 프레임 위치에서 오디오 클립을 재생한다. 처음 실행될 때 프레임 위치는 0

```
void stop()
```

재생중인 오디오 클립의 재생을 중단한다.

```
void setFramePosition(int frames)
```

오디오 샘플 내에 재생할 프레임의 위치를 지정한다. 프레임의 시작 번호는 0이다.

```
void loop(int count)
```

현재 프레임의 위치에서 시작하여 count만큼 반복 재생한다. count 값이 LOOP\_CONTINUOUSLY 이면 무한 반복한다.

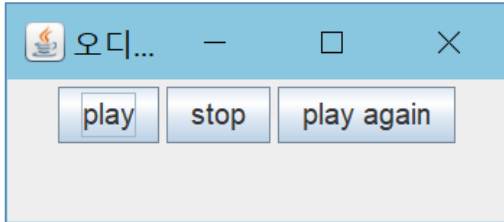
```
void close()
```

오디오 클립의 모든 자원을 반환한다.

# 예제 14-12 : 오디오 재생/중지

36

play, stop, play again의 3 버튼을 가진 응용프로그램을 작성하고, 애국가.wav 파일을 연주하라.  
애국가.wav는 프로젝트 폴더 밑의 audio 폴더에 있어야 한다.



```
import javax.sound.sampled.*;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.io.*;

public class AudioEx extends JFrame {
    private JButton btns[] = { new JButton("play"),
                               new JButton("stop"), new JButton("play again")};
    private Clip clip;
    public AudioEx() {
        setTitle("오디오 제어");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300,200);
        Container c = getContentPane();
        c.setLayout(new FlowLayout());
        MyActionListener al = new MyActionListener();
        for(int i=0; i<btns.length; i++) {
            c.add(btns[i]);
            btns[i].addActionListener(al);
        }
        setVisible(true);
        loadAudio("audio/애국가1절.wav");
    }
}
```

```
private void loadAudio(String pathName) {
    try {
        clip = AudioSystem.getClip(); // 비어있는 오디오 클립 만들기
        File audioFile = new File(pathName); // 오디오 파일의 경로명
        AudioInputStream audioStream =
            AudioSystem.getAudioInputStream(audioFile);
        clip.open(audioStream); // 재생할 오디오 스트림 열기
    }
    catch (LineUnavailableException e) { e.printStackTrace(); }
    catch (UnsupportedAudioFileException e) { e.printStackTrace(); }
    catch (IOException e) { e.printStackTrace(); }
}

class MyActionListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        switch(e.getActionCommand()) {
            case "play": clip.start(); // 오디오 재생 시작
                        break;
            case "stop": clip.stop(); // 오디오 재생 중단
                        break;
            case "play again": clip.setPosition(0);
                               clip.start();
                               break;
        }
    }
}

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

# 오디오 클립에 Line 이벤트 처리

37

## □ Line 이벤트

- 오디오 재생 과정에서 오디오 클립(Clip 객체)에 발생하는 이벤트
  - 오디오가 재생을 시작할 때
  - 재생이 중단되었을 때
  - 오디오 클립이 닫혔을 때

## □ LineListener 인터페이스

```
public void update(LineEvent e)
```

오디오가 재생을 시작할 때, 재생이 중단되었을 때, 오디오 클립이 닫혔을 때 호출된다.

## □ LineEvent 객체

```
Line getLine()
```

이벤트가 발생하는 오디오 클립 객체(Clip은 Line 인터페이스 상속받음)

```
Long getFramePosition()
```

오디오 내에 이벤트가 발생한 프레임 위치(0부터 시작)

```
LineEvent.Type getType()
```

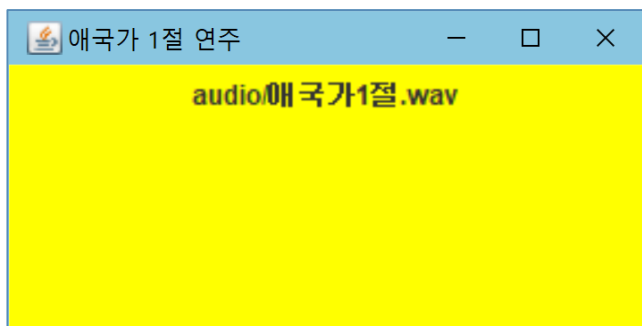
이벤트 종류로 다음 4가지 값 중 하나 리턴

- LineEvent.Type.OPEN: 오디오 클립이 열릴 때(Clip의 open() 호출시)
- LineEvent.Type.START: 재생이 시작될 때(Clip의 start() 호출시)
- LineEvent.Type.STOP: 재생이 중단될 때(Clip의 stop()이나 끝까지 재생되었을 때)
- LineEvent.Type.CLOSE: 오디오 클립이 닫히고 모든 자원이 반환되었을 때(Clip의 close() 호출시)

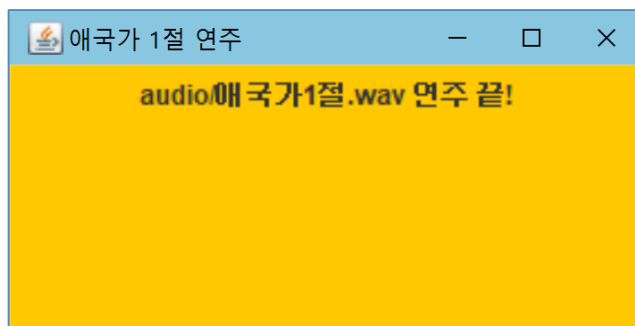
# 예제 14-13 : 오디오 연주가 끝나면 콘텐츠팬의 배경색을 ORANGE 색으로 만들기

38

애국가1절.wav 파일을 재생시키고 연주가 끝나면 콘텐츠팬의 바탕색이 ORANGE 색으로 바뀌도록 하라.



애국가의 연주 중



애국가의 연주가 완전히 끝났을 때

# 예제 14-13 : 정답 코드

39

```
import javax.sound.sampled.*;
import javax.swing.*;
import java.awt.*;
import java.io.*;

public class LineEventEx extends JFrame {
    private Clip clip;
    private String song="audio/애국가1절.wav";
    private JLabel label=new JLabel(song);

    public LineEventEx() {
        setTitle("애국가 1절 연주");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();

        c.setLayout(new FlowLayout());
        c.setBackground(Color.YELLOW);
        c.add(label);

        setSize(300,150);
        setVisible(true);

        loadAudio(song);
    }
}
```

```
private void loadAudio(String pathName) {
    try {
        File audioFile = new File(pathName);
        final AudioInputStream audioStream =
            AudioSystem.getAudioInputStream(audioFile);
        clip = AudioSystem.getClip();

        clip.addLineListener(new LineListener() {
            public void update(LineEvent e) {
                if (e.getType() == LineEvent.Type.STOP) {
                    try {
                        getContentPane().setBackground(Color.ORANGE);
                        label.setText(song + " 연주 끝!");
                        audioStream.close();
                    } catch (IOException e1) {
                        e1.printStackTrace();
                    }
                }
            }
        });
        clip.open(audioStream);
        clip.start();
    }
    catch (LineUnavailableException e) { e.printStackTrace(); }
    catch (UnsupportedAudioFileException e) { e.printStackTrace(); }
    catch (IOException e) { e.printStackTrace(); }
}

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