

(INTERMEDIATE) JAVA PROGRAMMING

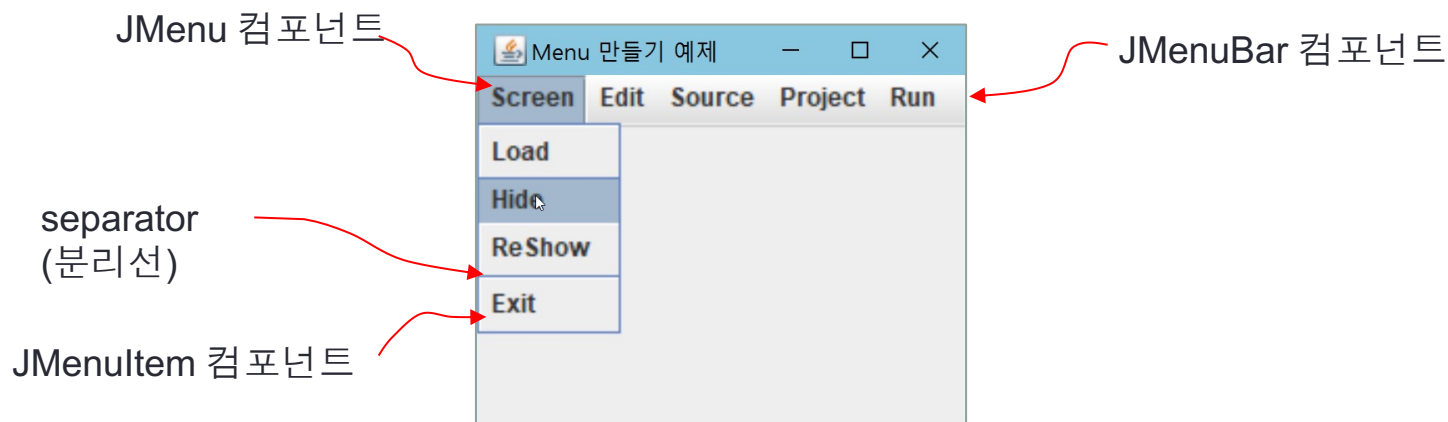
More Swing Components
Chapter 14

JAVA:

MORE SWING COMPONENTS

메뉴 만들기

- 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

JMenuBar 컴포넌트



JMenuBar 컴포넌트



add



JMenu 컴포넌트

JMenuBar 컴포넌트



add



JMenuItem 컴포넌트

JMenu 컴포넌트

JMenuBar 컴포넌트



JMenu 컴포넌트

JMenu 컴포넌트

(1) JMenuBar 컴포넌트 생성

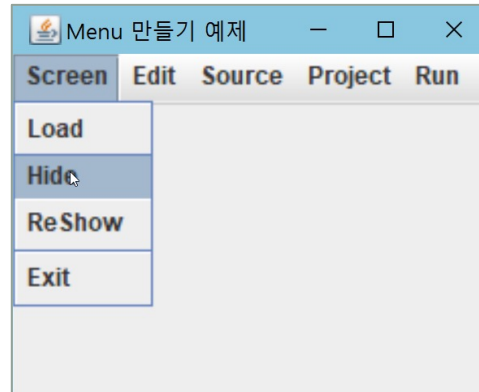
(2) JMenu 컴포넌트를
생성하여 JMenuBar에
붙인다.

(3) JMenuItem 컴포넌트를
생성하여 JMenu에
붙인다.

(3') 여러 개의 메뉴와
메뉴 아이টে를 생성한다.



(4) JMenuBar 컴포넌트를
JFrame에 붙인다.



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

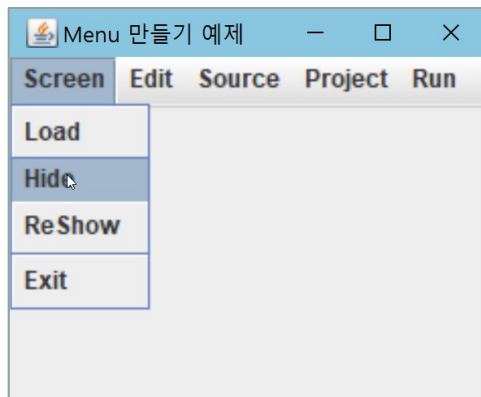
JMenu sMenu = new JMenu("Screen");
(2) → mb.add(sMenu);

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

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"));

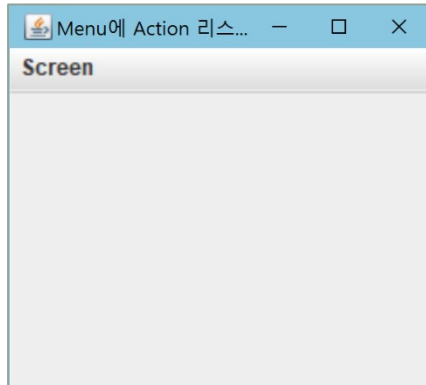
        setJMenuBar(mb);
    }
    public static void main(String [] args) {
        new MenuEx();
    }
}
```

메뉴아이템에 Action 이벤트 달기

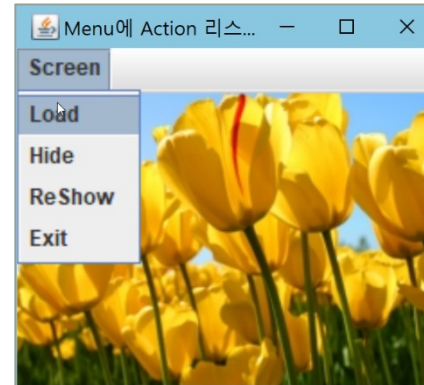
- ActionListener 이용
 - 각 JMenuItem 컴포넌트에 Action 이벤트 리스너 설정
 - 메뉴 아이템이 선택되었을 때 Action 이벤트 리스너 호출
- 메뉴 아이템에 ActionListener를 다는 예

```
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  
    }  
}
```

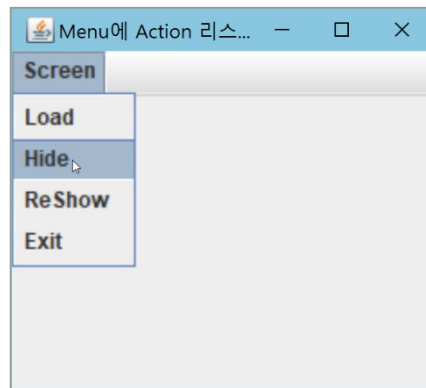
예제 14-2 : 메뉴아이템 선택 시 Action 리스너 활용



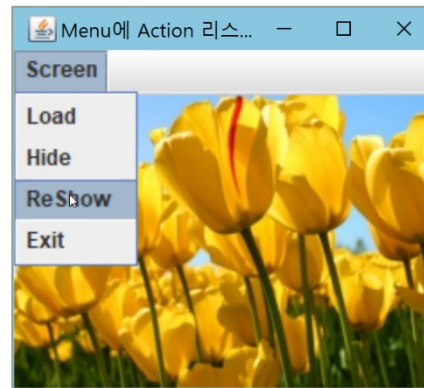
초기상태



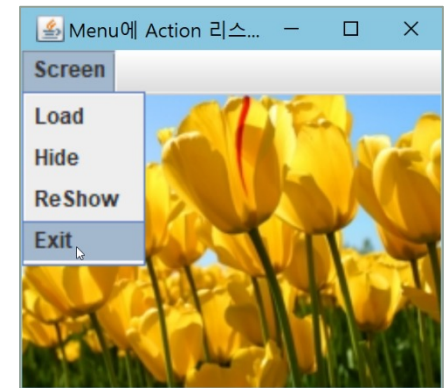
Load 메뉴아이템 선택하면
레이블을 이용하여 이미지 출력



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

```
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);
    }
}
```

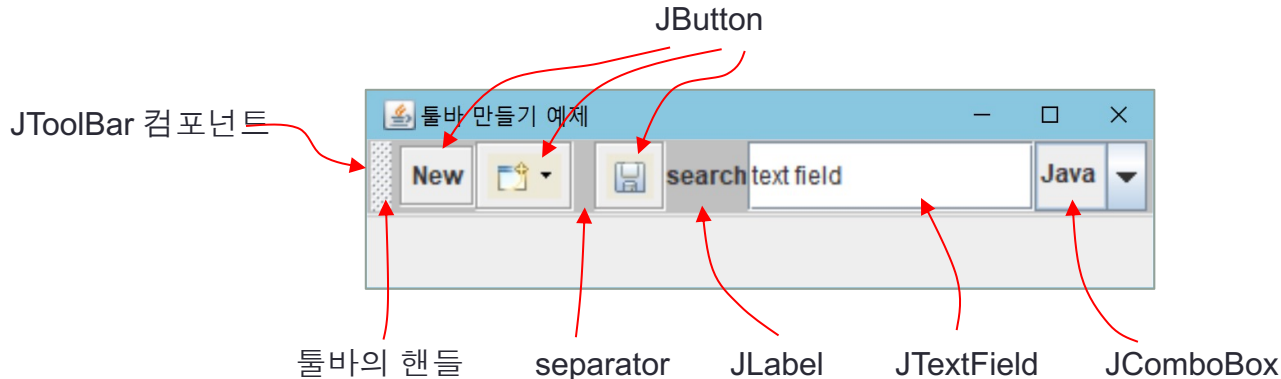
```
class MenuActionListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        String cmd = e.getActionCommand();
        switch(cmd) { // 메뉴 아이템의 종류 구분
            case "Load" :
                if(imgLabel.getIcon() != null)
                    return; // 이미 로딩되었으면 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

• JToolBar

- 투바를 구현한 컴포넌트
 - 여러 컴포넌트를 담을 수 있는 컨테이너
 - 버튼이나 이미지 등 모든 컴포넌트를 부착하여 이들을 메뉴처럼 사용
- 투바의 모양
 - 한 행 혹은 한 열로만 표현
- 투바가 부착되는 위치
 - 투바는 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 만들기 와 제어

- 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

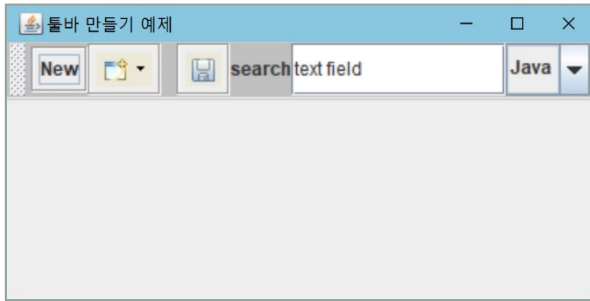
- 툴바 드래그 못하게 고정하는 방법

- JToolBar.setFloatable(boolean b)
 - b가 true이면 드래그하여 이동 가능한 툴바, false이면 툴바의 핸들이 없어지며 드래그 불가능(이동 불가능)

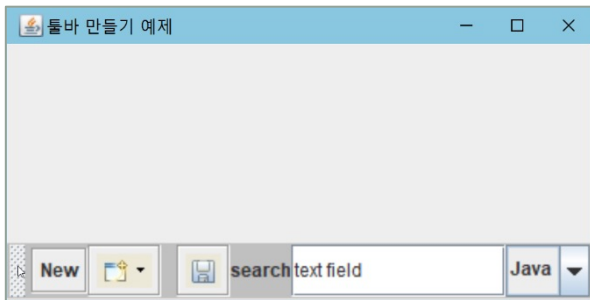
- 툴바에 separator 삽입

- JToolBar.addSeparator()

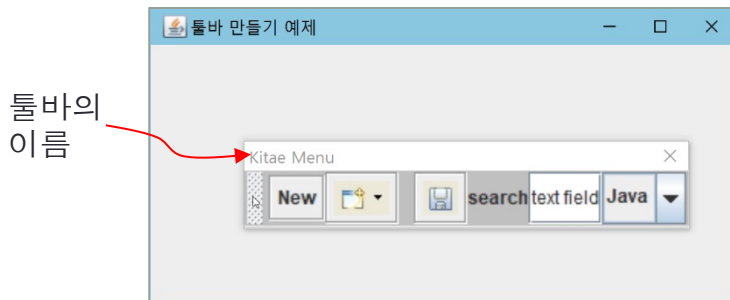
예제 14-3 : 툴바 만들기



초기 컨테트팬의 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);
```

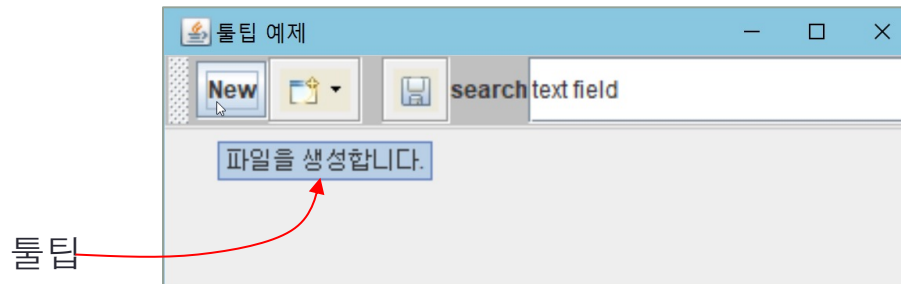
```
    }
```

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

툴팁

• 툴팁

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

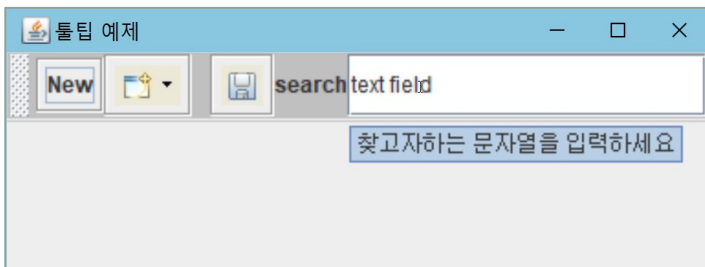
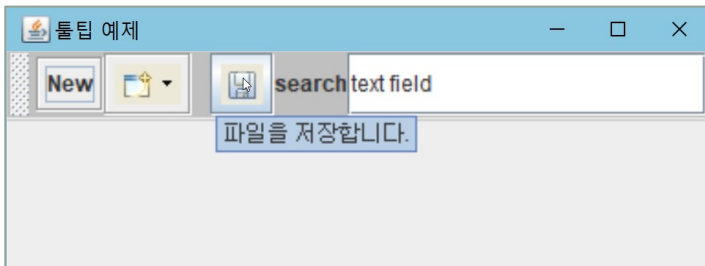
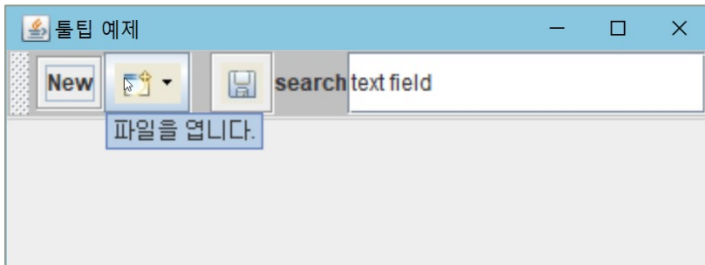
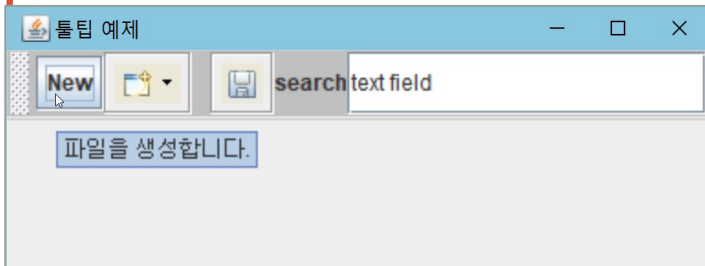


• 툴팁 생성

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

- 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 올리자마자 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);

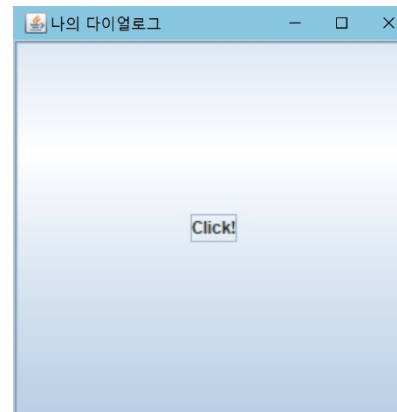
        setSize(400,200);
        setVisible(true);
    }

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

다이얼로그 만들기

- 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

17

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

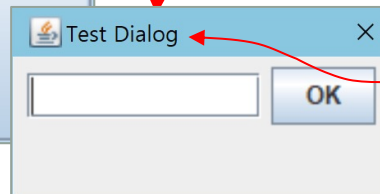
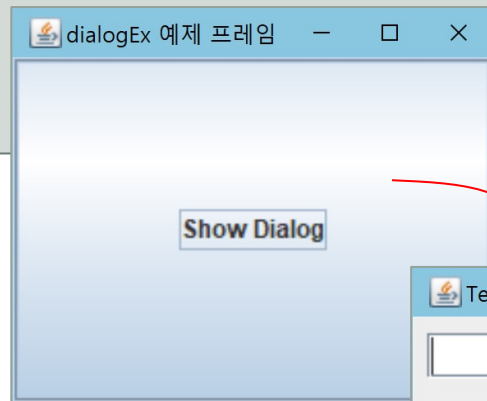
class MyDialog extends JDialog {
    private JTextField tf = new JTextField(10);
    private JButton okButton = new JButton("OK");

    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

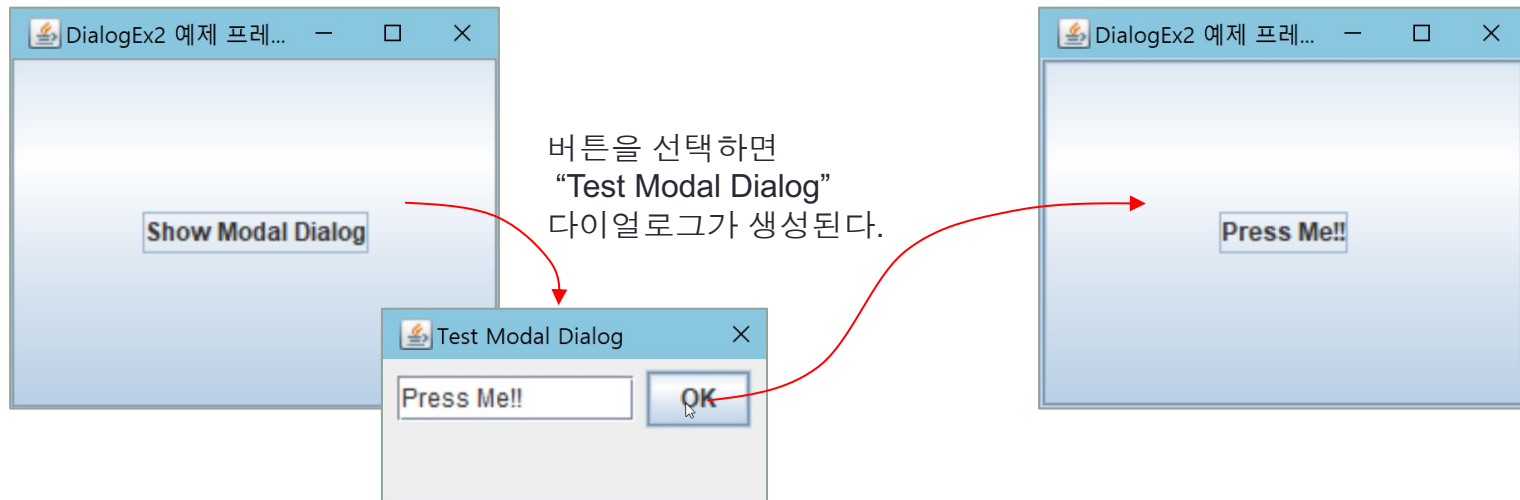
MyDialog로 만든 다이얼로그

모달 다이얼로그와 모달리스 다이얼로그

- 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

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();

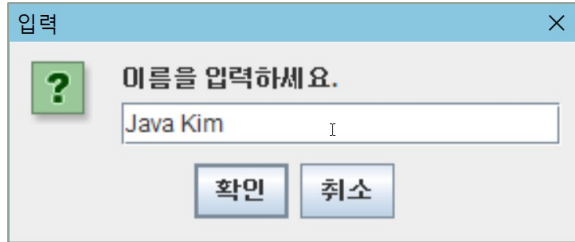
                if(text == null) return;
                JButton btn = (JButton)e.getSource();
                btn.setText(text);
            }
        });
        getContentPane().add(btn);
        setSize(250, 200);
        setVisible(true);
    }
    public static void main(String[] args) {
        new DialogEx2();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

팝업 다이얼로그, JOptionPane

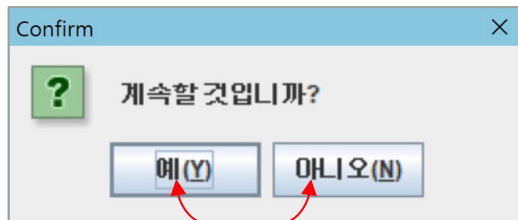
- JOptionPane
 - 간단하고 유용한 팝업 다이얼로그 지원
 - static 의 타입의 간단한 메소드 이용
- 입력 다이얼로그
 - 한 줄을 입력 받는 다이얼로그
 - `String JOptionPane.showInputDialog(String msg)`
- 확인 다이얼로그
 - 사용자로부터 Yes/No 등 확인을 입력받는 다이얼로그
 - `int JOptionPane.showConfirmDialog(Component parentComponent, Object msg, String title, int optionType)`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

JOptionPane 팝업 다이얼로그와 코드 샘플

22

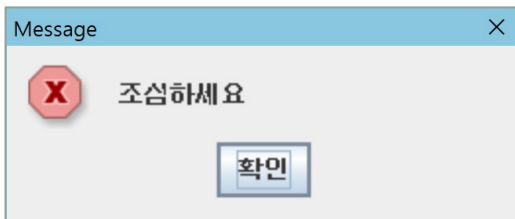


```
String name = JOptionPane.showInputDialog("이름을 입력하세요.");  
// name에 "Java Kim" 리턴  
// 사용자가 입력 없이 창을 닫으면 null 리턴
```



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);
    }
});

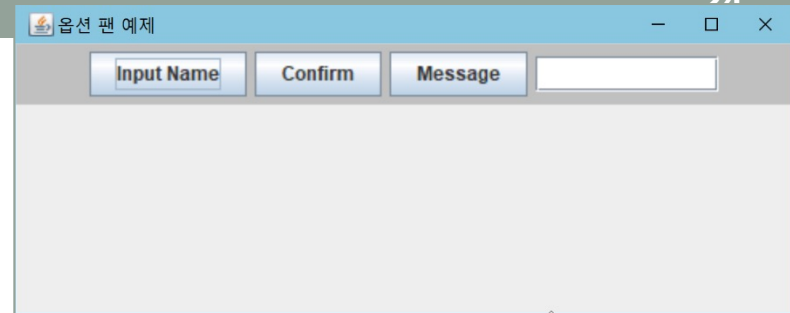
confirmBtn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        int result = JOptionPane.showConfirmDialog(null,
            "계속할 것입니까?", "Confirm",
            JOptionPane.YES_NO_OPTION);
        if(result == JOptionPane.CLOSED_OPTION)
            tf.setText("Just 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);
    }
});

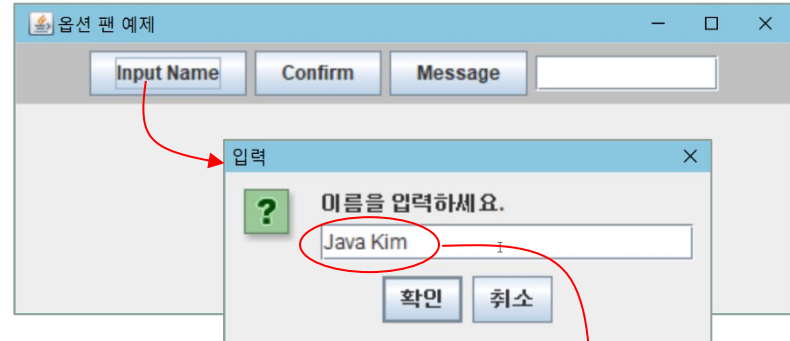
public static void main(String [] args) {
    new OptionPaneEx();
}
```

예제 실행

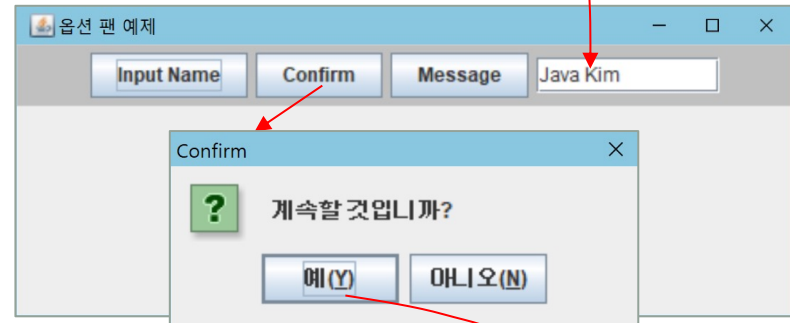
초기 화면



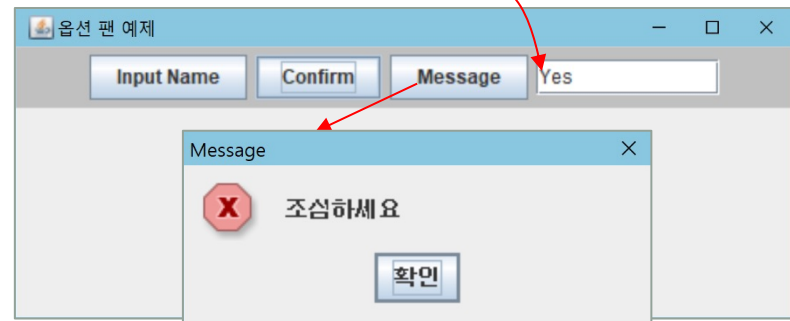
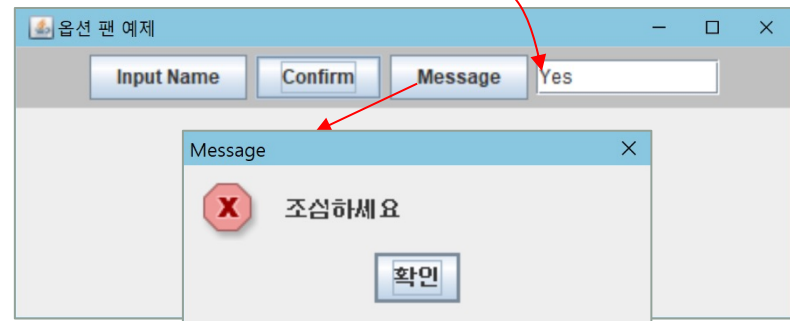
Input Name 버튼을 누르는 경우 입력 다이얼로그 생성
"Java Kim"을 입력하고 확인 버튼을 누르면
텍스트필드 창에 출력



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

- 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

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

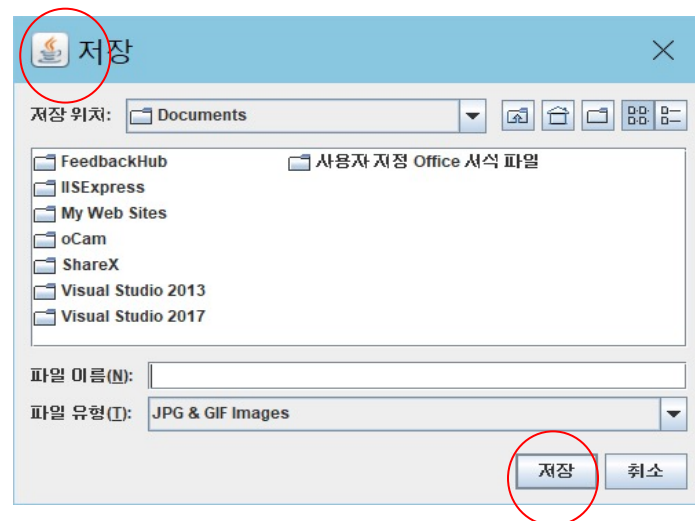
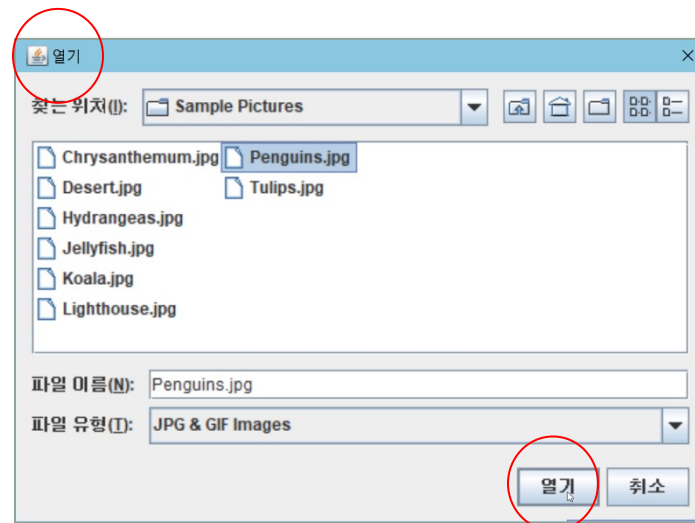
생략가능

4. 열기 다이얼로그 출력

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

5. 사용자가 선택한 파일 이름 알아내기

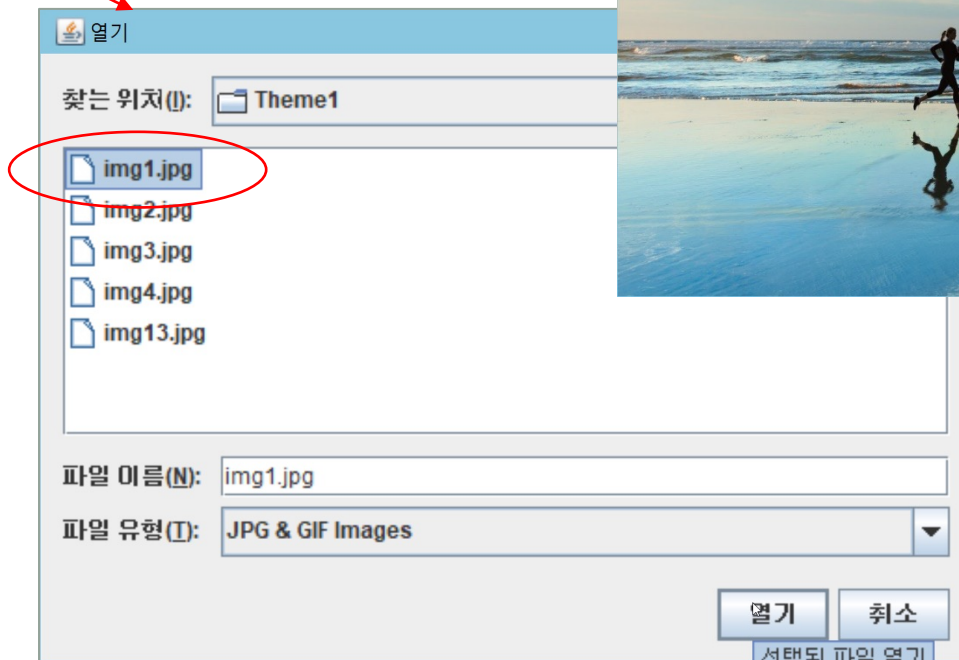
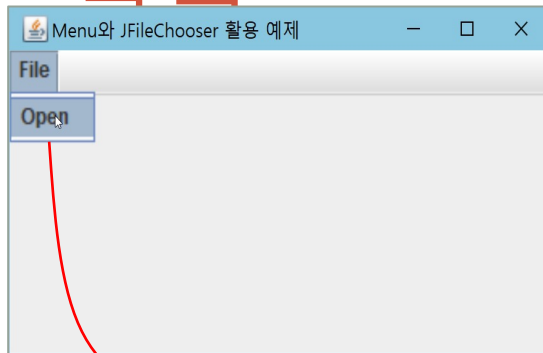
```
if(ret == JFileChooser.APPROVE_OPTION) {
    String pathName = chooser.getSelectedFile().getPath();
    String fileName = chooser.getSelectedFile().getName();
}
```



파일 저장 다이얼로그 생성 과정

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

예제 14-9 : 파일 열기 다이얼로그 생성 및 축련



예제 14-9 : 정답 코드

```
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

- JColorChooser
 - 색상 팔레트를 제공하는 모달 다이얼로그

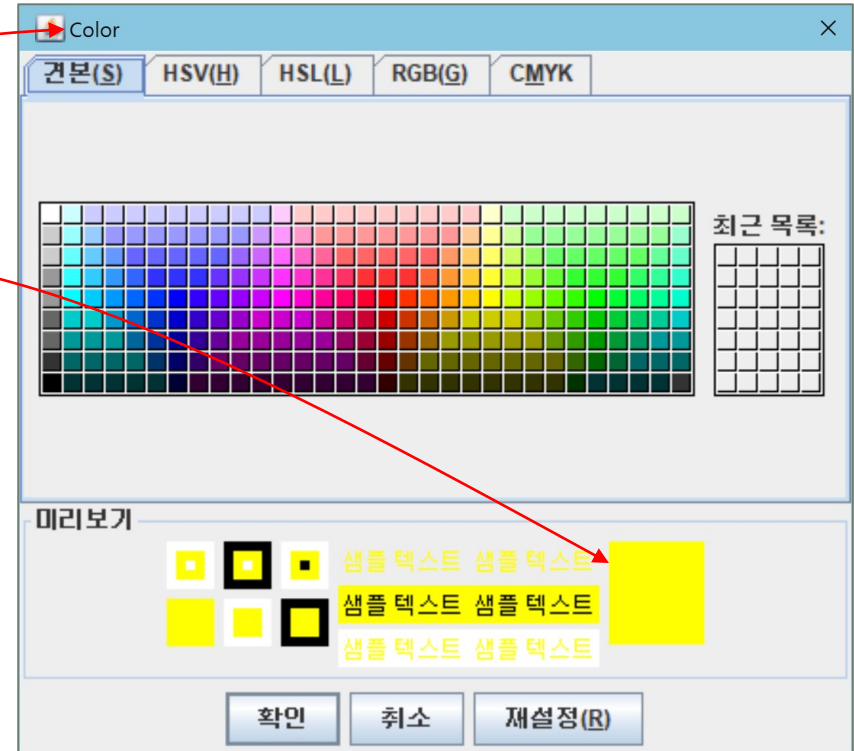
컬러 다이얼로그 출력 과정

1. 컬러 다이얼로그 출력

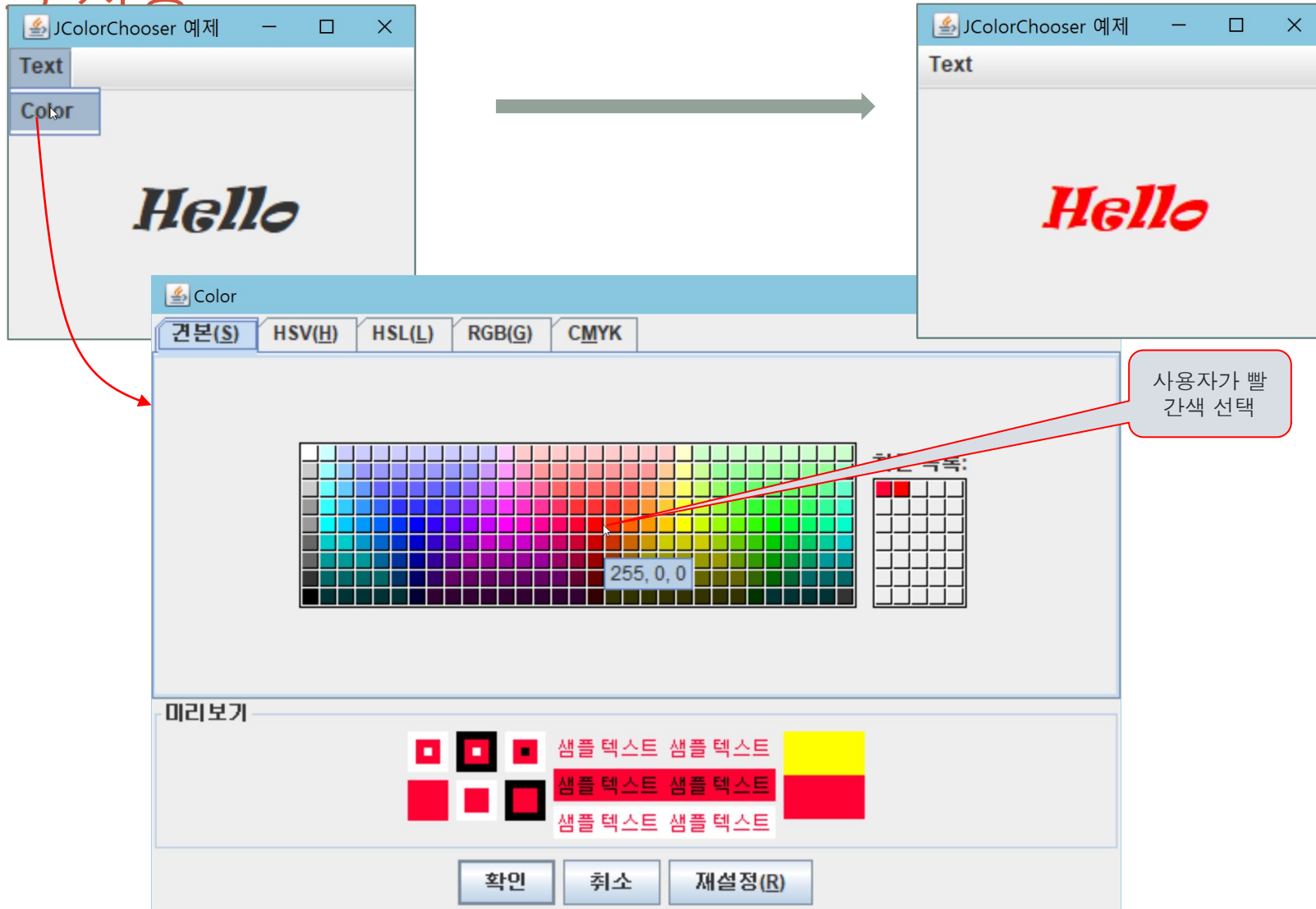
```
Color selectedColor = JColorChooser.showDialog(null,
    "Color",
    Color.YELLOW);
// 사용자가 "확인" 버튼을 누르면 선택한 색을 리턴하지만
// 사용자가 "취소" 버튼이나 다이얼로그를 강제로 닫을 때
// null 리턴
```

3. 사용자가 선택한 색 알아내기

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



예제 14-10 : JColorChooser를 이용한 컬러 다이얼로 그 사용



예제 14-10 : 정답 코드

```
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();
}
}
```