

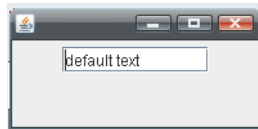


Chapter 10

02204271

Event Handling and GUI

Text Input



- ใช้สำหรับให้ผู้ใช้กรอกข้อมูลลงใน text box หรือ text field
- component: JTextField
 - JTextField(int cols): cols บอกขนาดของ text box ที่ต้องการ

```
JTextField(String text, int cols): text เป็นค่าเริ่มต้นใน text box
JFrame f = new JFrame();
f.setSize(100, 100);
JPanel pane = new JPanel();
JTextField textField = new JTextField("default text", 10);
pane.add(textField);
f.add(pane);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
f.setVisible(true);
```

Graphical User Components



- top-level container
 - frame หรือ main window □ JFrame
 - นอกจาก JFrame ยังมี JDialog, JApplet
 - มีหน้าที่ไว้ให้ component อื่นๆ มาวาง
- intermediate container
 - มีเพื่อให้ง่ายต่อการวางตำแหน่งของ component อื่นๆ
 - เช่น JPanel, JScrollPane, JTabbedPane เป็นต้น
- atomic components
 - องค์ประกอบที่มีข้อมูลและมีความหมายในตัว
 - เช่น JTextField, JComboBox, JLabel, JTable, JButton

Label



- label เป็นข้อความธรรมดาบน GUI และไม่มีการตอบสนองใดกับผู้ใช้
- ใช้สำหรับกำกับ component อื่นๆ
- component: JLabel
 - JLabel(String text)
 - JLabel(Icon icon)

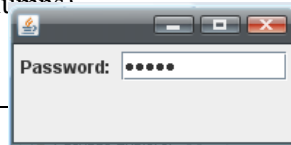


```
JLabel(String text, int align);
JLabel(Icon icon, int align);
JLabel(String text, Icon icon, int align);
JPanel pane = new JPanel();
JLabel label = new JLabel("Name", new ImageIcon("info.png"), JLabel.LEFT);
JTextField textField = new JTextField(10);
pane.add(label);
pane.add(textField);
f.add(pane);
```

Password Field



- เป็น text field ชนิดหนึ่งที่ใช้สำหรับกรอก password เพื่อไม่ให้เห็นตัวอักษรที่ป้อน
- component: JPasswordField
- JPasswordField(String text, int columns)
- JPasswordField(int columns)
- JPasswordField(String text)



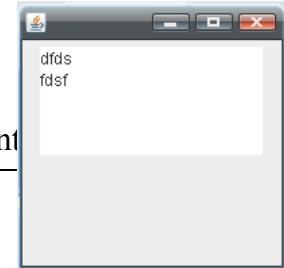
```
JFrame f = new JFrame();
f.setSize(200, 100);
JPanel pane = new JPanel();
JLabel label = new JLabel("Password: ", JLabel.RIGHT);
JPasswordField pass = new JPasswordField(10);
pane.add(label);
pane.add(pass);
f.add(pane);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
f.setVisible(true);
```

Kasetsart University

Text Area



- ใช้สำหรับรับ input จากผู้ใช้มากกว่า 1 บรรทัด
- component: JTextArea
- JTextArea()
- JTextArea(int lines, int columns)
- JTextArea(String text, int lines, int columns)



```
JFrame f = new JFrame();
f.setSize(200, 200);
JPanel pane = new JPanel();
JTextArea area = new JTextArea(5, 15);
pane.add(area);

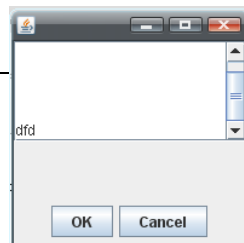
f.add(pane);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
f.setVisible(true);
```

Kasetsart University

Scroll pane



- เป็น container ที่มี scroll bar ใช้ในกรณีที่ข้อมูลเกินพื้นที่ที่มีให้
- component: JScrollPane



```
JFrame f = new JFrame();
f.setSize(200, 200);
JTextArea area = new JTextArea(5, 15);
JScrollPane pane = new JScrollPane(area);
```

```
JPanel pane2 = new JPanel();
JButton b1 = new JButton("OK");
JButton b2 = new JButton("Cancel");
pane2.add(b1);
pane2.add(b2);
```

```
f.add(pane, BorderLayout.NORTH);
f.add(pane2, BorderLayout.SOUTH);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
f.setVisible(true);
```

Kasetsart University

Checkbox



- เป็น component สำหรับให้ติ๊ก ซึ่งผู้ใช้สามารถเลือกหรือไม่เลือก checkbox หนึ่งๆ ก็ได้
- component: JCheckBox
- JCheckBox(String label)
- JCheckBox(String label, Icon icon)
- JCheckBox(String label, boolean state)

```
JPanel pane2 = new JPanel();
JLabel txt = new JLabel("ผลไม้ที่ชอบ : ");
pane2.add(txt);
JPanel pane = new JPanel();
pane.setLayout(new GridLayout(0, 2));
final JLabel pic1 = new JLabel();
final JLabel pic2 = new JLabel();
final JCheckBox box1 = new JCheckBox("แอปเปิ้ล");
final JCheckBox box2 = new JCheckBox("ส้ม");
```

Kasetsart University

Checkbox

```
ActionListener listener = new ActionListener(){
    public void actionPerformed(ActionEvent e) {
        ImageIcon img1 = box1.isSelected()?
            new ImageIcon("apple.jpg"):null;
        pic1.setIcon(img1);
        ImageIcon img2 = box2.isSelected()?
            new ImageIcon("orange.jpg"):null;
        pic2.setIcon(img2);
    }
};

box1.addActionListener(listener);
box2.addActionListener(listener);
pane.add(box1);
pane.add(pic1);
pane.add(box2);
pane.add(pic2);
f.add(pane, BorderLayout.NORTH);
f.add(pane);
```

Kasetsart University



radio button

- เป็น component สำหรับให้เลือกเอาอันใดอันหนึ่งจากหลายอัน
- component: JRadioButton
 - JRadioButton(String label)
 - JRadioButton(String label, Icon icon)
 - JRadioButton(String label, boolean state)

• จะต้องมี ButtonGroup เพื่อให้ในการจัดกลุ่มให้กับ radio buttons

```
public class RadioFrame extends JFrame {
    private ButtonGroup group = new ButtonGroup();
    private JPanel pane = new JPanel();
    JLabel txt;

    ... // ต่ออีกหน้า

    public static void main(String[] args) {
        RadioFrame f = new RadioFrame();
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        f.setVisible(true);
    }
}
```

Kasetsart University



radio button

```
public RadioFrame() {
    setSize(400,120);
    JPanel pane2 = new JPanel();
    txt = new JLabel("The brown fox jumps over the lazy dog.");
    pane2.add(txt);
    addRadioButton("small",10);
    addRadioButton("medium",16);
    addRadioButton("large",24);
    add(pane2);
    add(pane, BorderLayout.SOUTH);
}

public void addRadioButton(String name,final int fontSize) {
    JRadioButton radio = new JRadioButton(name);
    group.add(radio);
    pane.add(radio);
    ActionListener listener = new ActionListener(){
        public void actionPerformed(ActionEvent e) {
            txt.setFont(new Font("Serif", Font.PLAIN, fontSize));
        }
    };
    radio.addActionListener(listener);
}
```

Kasetsart University



Combo Box

- คือ drop-down list ไว้สำหรับเลือกรายการหนึ่งจากหลายรายการ
 - ใช้ในกรณีที่รายการมีมาก (>4) ซึ่งไม่เหมาะที่จะใช้ radio button

- component: JComboBox

```
JFrame f = new JFrame();
JPanel pane2 = new JPanel();
final JLabel txt = new JLabel("The brown fox jumps over the lazy dog.");
pane2.add(txt);

final JComboBox combo = new JComboBox();
combo.setEditable(true);
combo.addItem("lilly");
combo.addItem("SansSerif");
combo.addItem("Monospaced");
combo.addItem("Dialog");
combo.addItem("Times New Roman");
```

Kasetsart University



Combo Box

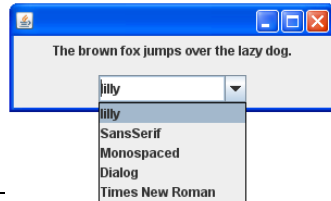


```

combo.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent event) {
        txt.setFont(new Font((String) combo.getSelectedItem(),
            Font.PLAIN, 16));
    }
});

pane.add(combo);

f.add(pane2);
f.add(pane, BorderLayout.SOUTH);
    
```



Kasetsart University

Slider



- combo box ใช้เลือกค่าแบบ discrete แต่ slider ทำให้เราเลือกค่าแบบ continue
- component: JSlider
 - JSlider(int min, int max, int initialValue)

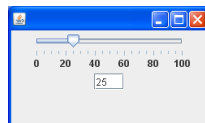
Kasetsart University

Slider



```

JPanel pane = new JPanel();
final JTextField textField = new JTextField("25",3);
JSlider slider = new JSlider(0,100,25);
slider.setPaintTicks(true);
slider.setPaintLabels(true);
slider.setMajorTickSpacing(20);
slider.setMinorTickSpacing(5);
slider.addChangeListener(new ChangeListener() {
    public void stateChanged(ChangeEvent e) {
        JSlider source = (JSlider) e.getSource();
        textField.setText("" + source.getValue());
    }
});
pane.add(slider);
pane.add(textField);
    
```



Kasetsart University

Menu



- เรามักต้องการสร้าง menu บน

- components:

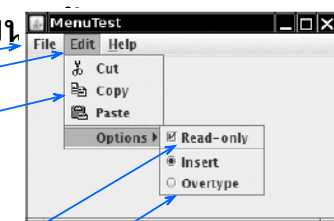
- JMenuBar
- JMenu
- JMenuItem

- JMenuItem(Action a):

menu item พร้อมด้วยการกระทำ

- JMenuItem(String label)
- JMenuItem(String label, Icon icon)

- JCheckBoxMenuItem

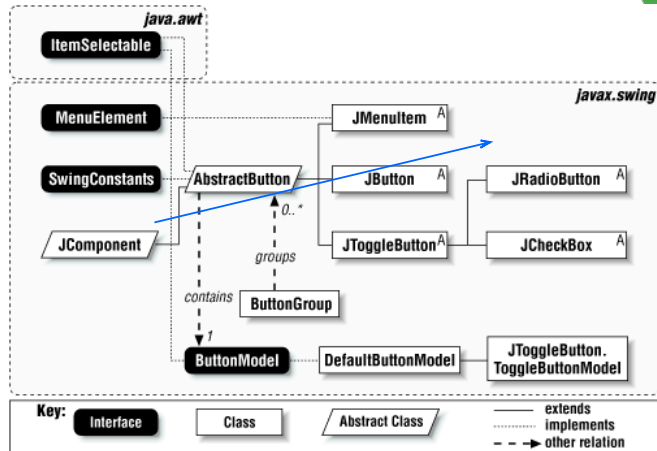


AbstractButton

JMenuItem

Kasetsart University

Menu



Kasetsart University

Menu

```

JMenuBar menuBar = new JMenuBar();
f.setJMenuBar(menuBar);
JMenu editMenu = new JMenu("Edit");
menuBar.add(editMenu);
JMenuItem pasteItem = new JMenuItem("Paste");
editMenu.add(pasteItem);
editMenu.addSeparator();
pasteItem.addActionListener(listener);
JMenu optionsMenu = new JMenu("Options"); // a submenu
editMenu.add(optionsMenu);
JCheckBoxMenuItem readonlyItem = new
JCheckBoxMenuItem("Read-only");
optionsMenu.add(readonlyItem);
Action exitAction = new AbstractAction("Exit") {
    public void actionPerformed(ActionEvent event) {
        System.exit(0);
    }
};
editMenu.add(exitAction);
    
```

Kasetsart University

dialog box

- dialog box เป็นเหมือน pop-up window ที่มักจะแสดงชั่วคราวเพื่อทำอะไรสักอย่างกับหน้าต่างโปรแกรม
- แบ่งเป็น 2 ประเภทใหญ่ๆ
 - modal dialog box: dialog box ที่เมื่อแสดงออกมาแล้ว จะไม่ให้ผู้ใช้ทำอะไรกับหน้าต่างหลักได้จนกว่า dialog box จะปิดไป
 - modeless dialog box: dialog box ที่ผู้ใช้สามารถใช้งานไปพร้อมๆ กับหน้าต่างหลักได้ เช่น toolbar

dialog box

- JOptionPane มี dialog พื้นฐาน 4 แบบ ซึ่งเรียกใช้ผ่าน static methods
 - showMessageDialog: แสดง dialog โดยมีปุ่ม OK รอให้ผู้ใช้กด
 - showConfirmDialog: แสดง dialog โดยมีปุ่มให้เลือก 2 ปุ่ม (OK/Cancel)
 - showOptionDialog: แสดง dialog เพื่อให้ผู้ใช้เลือกตัวเลือกหนึ่งจากตัวเลือกที่มีให้
 - showInputDialog: แสดง dialog โดยให้ผู้ใช้ป้อนข้อมูล

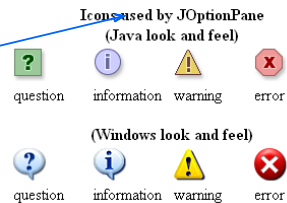
dialog box



- dialog ทั้ง 4 แบบนี้ประกอบด้วย components 3 ตัว

- icon

- ERROR_MESSAGE
- INFORMATION_MESSAGE
- WARNING_MESSAGE
- QUESTION_MESSAGE
- PLAIN_MESSAGE



Kasetsart University

dialog box



- ข้อความ

- String
- Icon
- Component: user interface component
- Object[]: แสดงวัตถุทุกอันใน array
- วัตถุอื่นๆ : วัตถุอื่นๆ ในรูปแบบ string จาก toString()

- ปุ่มให้เลือกกด: ขึ้นกับชนิดของ dialog
 - confirm dialog: DEFAULT_OPTION, YES_NO_OPTION,

Kasetsart University

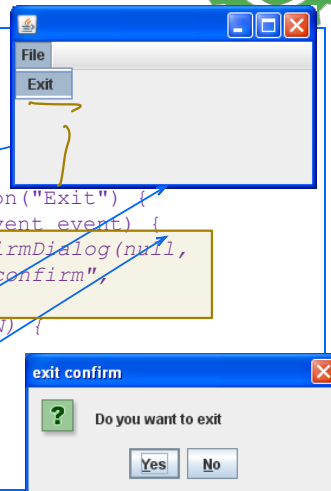
dialog box



```
JMenuBar menuBar = new JMenuBar();
f.setJMenuBar(menuBar);
```

```
JMenu fileMenu = new JMenu("File");
menuBar.add(fileMenu);
```

```
Action exitAction = new AbstractAction("Exit") {
    public void actionPerformed(ActionEvent event) {
        int answer = JOptionPane.showConfirmDialog(null,
            "Do you want to exit", "exit confirm",
            JOptionPane.YES_NO_OPTION);
        if (answer==JOptionPane.YES_OPTION) {
            System.exit(0);
        }
    }
};
fileMenu.add(exitAction);
```



Kasetsart University

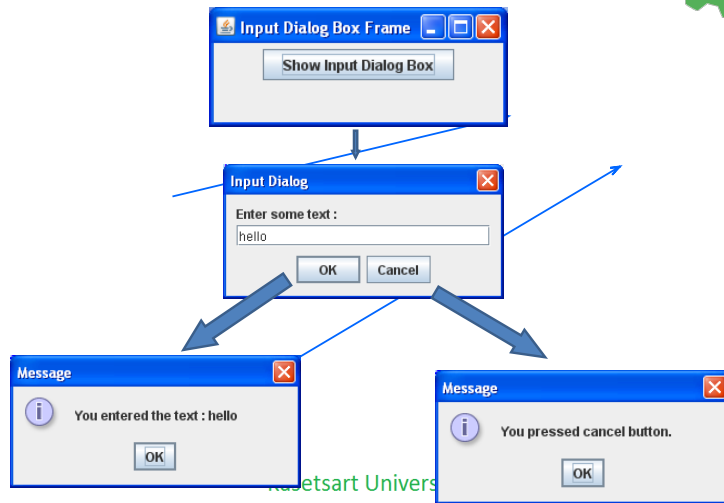
dialog box



```
JButton button = new JButton("Show Input Dialog Box");
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent ae) {
        String str = JOptionPane.showInputDialog(null,
            "Enter some text : ", "Input Dialog",
            JOptionPane.PLAIN_MESSAGE);
        if(str != null)
            JOptionPane.showMessageDialog(null,
                "You entered the text : " + str, "Message",
                JOptionPane.INFORMATION_MESSAGE);
        else
            JOptionPane.showMessageDialog(null,
                "You pressed cancel button.", "Message",
                JOptionPane.INFORMATION_MESSAGE);
    }
});
JPanel panel = new JPanel();
panel.add(button);
frame.add(panel);
```

Kasetsart University

dialog box



dialog box : custom



- สร้าง dialog ด้วยการ extends JDialog
- การใส่ component ต่างๆ เหมือนที่ทำใน JFrame

```
public class AboutDialog extends JDialog {
    public AboutDialog(JFrame parent, String message) {
        super(parent, "About Dialog", true); //true = modal
        JPanel messagePane = new JPanel();
        messagePane.add(new JLabel(message));

        JPanel buttonPane = new JPanel();
        JButton button = new JButton("OK");
        buttonPane.add(button);
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                setVisible(false);
                dispose();
            }
        });
    }
}
```

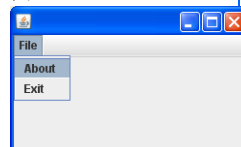
Kasetsart University

dialog box : custom



```
add(messagePane);
add(buttonPane, BorderLayout.SOUTH);
setDefaultCloseOperation(DISPOSE_ON_CLOSE);
pack();
setVisible(true);
}

public static void main(String[] args) {
    ...
    JMenuItem aboutItem = new JMenuItem("About");
    aboutItem.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent event) {
            new AboutDialog(f, "About Dialog test V. 1.0");
        }
    });
    fileMenu.add(aboutItem);
    ...
}
```

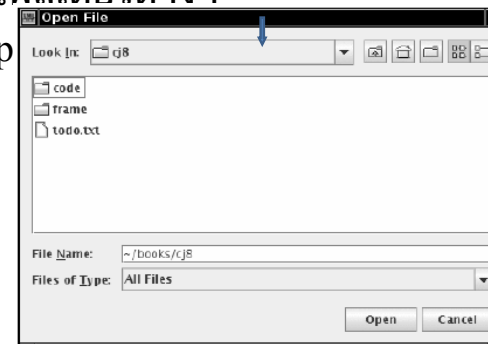


Kasetsart University

dialog box : File dialog



- File dialog ใช้สำหรับให้ผู้ใช้เลือกไฟล์ตาม path ของไฟล์เดอร์ต่างๆ
- Comp



Kasetsart University

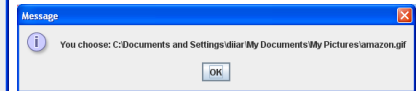
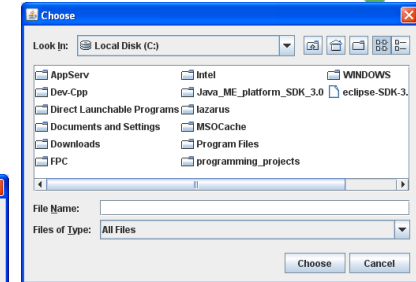
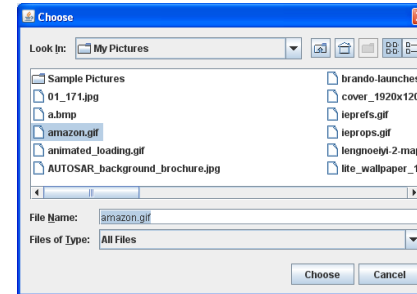
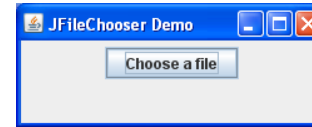
dialog box : File dialog



```
final JFrame frame = new JFrame("JFileChooser Demo");
final JFileChooser fc = new JFileChooser();
fc.setMultiSelectionEnabled(true);
fc.setCurrentDirectory(new File("C:\\"));
JButton btn1 = new JButton("Choose a file");
btn1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        int returnVal = fc.showDialog(frame, "Choose");
        if (returnVal == JFileChooser.APPROVE_OPTION) {
            File file = fc.getSelectedFile();
            JOptionPane.showMessageDialog(null, "You choose: "+
                file.getPath());
        }
    }
});
JPanel panel = new JPanel();
panel.add(btn1);
frame.add(panel);
frame.setSize(250, 100);
...
```

Kasetsart University

dialog box : File dialog



Kasetsart University



- <http://www.ntu.edu.sg/home/ehchua/programming/java>

Kasetsart University