

Week8

Programming Fundamentals II

1

Course Outline

- | | |
|-----------------------------|---------------------------|
| 1. P2J (Basic) | 8. Events + GUI |
| 2. P2J (Control structures) | 9. UI programming |
| 3. P2J (Collection types) | 10. Exceptions |
| 4. Classes and methods | 11. Testing and debugging |
| 5. Inheritance | 12. Generics |
| 6. Polymorphism | 13. Team project |
| 7. Interfaces | 14. Concurrency |

2

Course Outline

- | | |
|---------------------------|-------------------------|
| 8. Events + GUI | เสนอหัวข้อ Project |
| 9. UI programming | Lab7 |
| 10. Exceptions | Lab8 |
| 11. Testing and debugging | Lab9 |
| 12. Generics | Lab10 |
| 13. Team project | เสนอความคืบหน้า Project |
| 14. Concurrency | - |

Final Exam

Lab Exam

Present Project

3

Week 8 Event + GUI

1. GUI
2. Introduction GUI
3. Container
4. Simple Component
5. Text Component
6. Any Button
7. Event

4

GUI

Programming Fundamentals II

5

Introduction

A graphical user interface (GUI) presents a user-friendly mechanism for interacting with an application.

- Pronounced “GOO-ee”
- Gives an application a distinctive “look-and-feel.”
- Consistent, intuitive user-interface components give users a sense of familiarity
- Learn new applications more quickly and use them more productively.

6

Introduction

- Built from GUI components.
- User interacts via the mouse, the keyboard or another form of input, such as voice recognition.
- IDEs
 - Provide GUI design tools to specify a component’s size, location and other attributes in a visual manner by using the mouse, keyboard and drag-and-drop.
 - Generate the GUI code for you.
 - Greatly simplify creating GUIs, but each IDE has different capabilities and generates different code.

7

GUI Object

```
// Create a button with text OK
```

```
JButton jbtOK = new JButton("OK");
```

```
// Create a label with text "Enter your name: "
```

```
JLabel jlblName = new JLabel("Enter your name: ");
```

```
// Create a text field with text "Type Name Here"
```

```
JTextField jtfName = new JTextField("Type Name Here");
```

```
// Create a check box with text bold
```

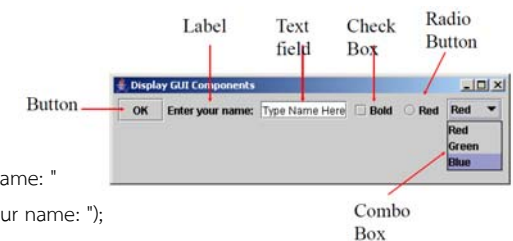
```
JCheckBox jchkBold = new JCheckBox("Bold");
```

```
// Create a radio button with text red
```

```
JRadioButton jrbRed = new JRadioButton("Red");
```

```
// Create a combo box with choices red, green, and blue
```

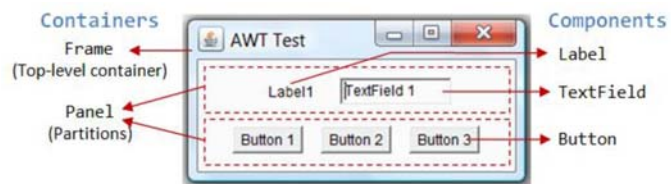
```
JComboBox jcoColor = new JComboBox(new String[]{"Red", "Green", "Blue"});
```



8

JAVA GUI

- Java AWT (Abstract Windowing Toolkit)
 - AWT Container คือที่ไว้รองรับ Component
 - AWT Component คือช่องรับข้อมูล
 - Layout Manager ไว้จัดตำแหน่งของ Component
- Java Swing



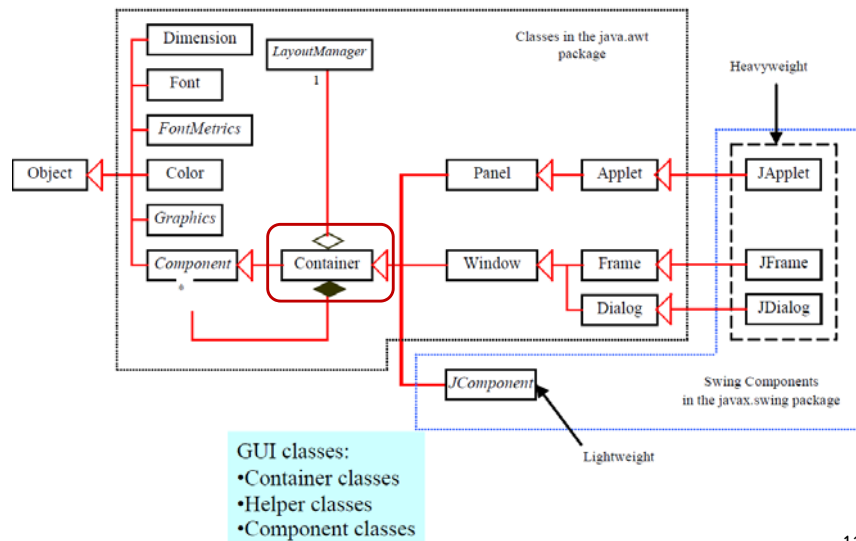
9

JAVA GUI

- Java Swing
 - เนื่องจากการที่ AWT เองมีปัญหาต่าง ๆ เกิดขึ้นในการใช้งาน เช่น เรื่องของความจำ
 - Java พัฒนา class ขึ้นมาใหม่ (Swing) ที่มีความสามารถในด้าน GUI แต่ปัญหาน้อยกว่า AWT
 - Swing ไม่ได้ใช้งานแทน AWT เพราะ AWT ยังมีใช้อยู่
 - Swing ก็ยังคงต้องนำคลาสจาก AWT มาใช้งาน

10

GUI Class Hierarchy (Swing)



11

Component AWT & Swing

AWT Comp.	Swing Comp.
Applet	Japplet
Component	Jcomponent
Container	-
Button	Jbutton
Canvas	-
CheckBox	JCheckBox
Dialog	Jdialog
Frame	Jframe
Label	JLabel

AWT Comp.	Swing Comp.
List	JList
Menu	JMenu
MenuBar	JMenuBar
MenuItem	JMenuItem
Panel	JPanel
ScrollBar	JScrollBar
TextArea	JTextArea
TextComponent	JTextComponent
TextField	JTextField

12

Swing Component Class

Swing Component class	Component
javax.swing.JButton	Button
javax.swing.JCheckBox	Checkbox
javax.swing.JDialog	Dialog
javax.swing.JFrame	Frame
javax.swing.JLabel	Label
javax.swing.JMenu	Menu
javax.swing.JPanel	Panel
javax.swing.JScrollbar	Scrollbar
javax.swing.JScrollPane	Scroll pane
javax.swing.JTextArea	Text Area
javax.swing.JTextField	Text Field
javax.swing.JComboBox	Combo Box
javax.swing.JList	List

13

Component

- Container
 - Top Level
 - Non - top Level
- Simple Component
- Text Component

14

Java Swing Component



JFrame



Swing Component

15

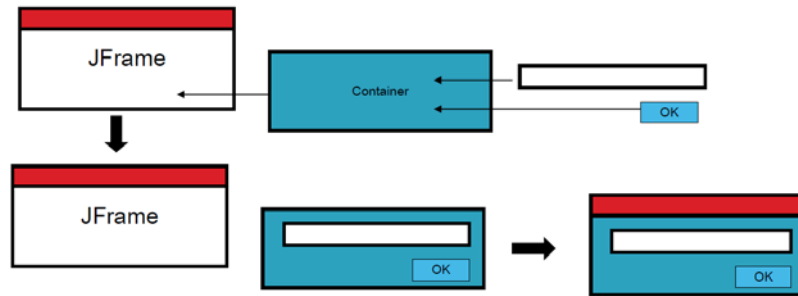
Frame

- Container
 - Top Level ←
 - Non - top Level
- Simple Component
- Text Component

16

Container

เนื่องจากวัตถุรับค่าต่างๆ ไม่สามารถอยู่อย่างอิสระบน JFrame ได้ จึงต้องใช้ Container ในการกำหนดตำแหน่งให้กับวัตถุรับค่าแต่ละตัว



17

Frame

- ประกอบด้วย (หัวเรื่อง, ขอบเขต, ขนาดพื้นที่ และปุ่มต่างๆ)
- GUI จะมี Frame อย่างน้อย 1 Frame เป็นองค์ประกอบเสมอ

รูปแบบ

```
JFrame frameName = new JFrame(title);
```

โดยที่ frameName เป็นชื่อ Object ที่ประกาศจาก Class JFrame
 title เป็นข้อความที่ต้องการให้แสดงบน Frame

18

Frame

```
3 import javax.swing.JFrame;
4
5 public class Lec01_FrameTest
6 {
7     public static void main(String[] args)
8     {
9         JFrame frame = new JFrame("FrameTest");
10    }
11 }
```

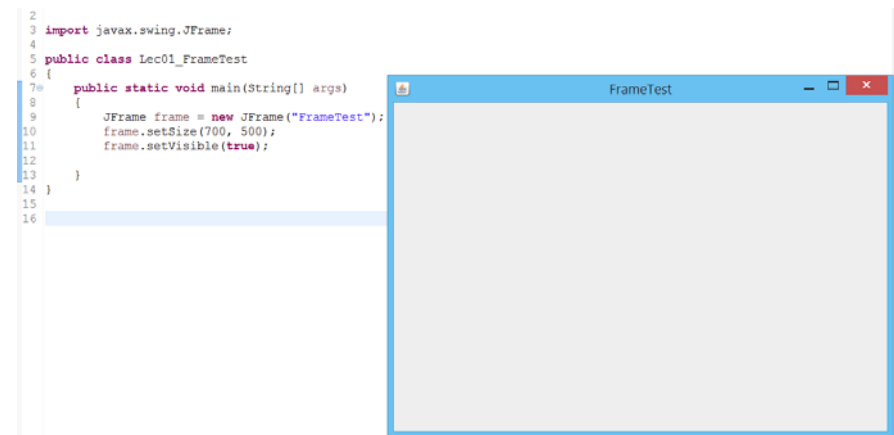
ทดลอง RUN!!!

จะไม่เห็นการทำงานของ JFrame

- Set Size
- Set Location
- Set Visible

19

Frame



20

JFrame Class

javax.swing.JFrame
<pre>+JFrame() +JFrame(title: String) +setSize(width: int, height: int): void +setLocation(x: int, y: int): void +setVisible(visible: boolean): void +setDefaultCloseOperation(mode: int): void +setLocationRelativeTo (c: Component): void</pre>

21

JFrame Class

- Method ที่สำคัญได้แก่
- public JFrame – ใช้สร้าง object แบบ JFrame
- public JFrame(String s) – ใช้สร้าง object แบบ JFrame โดยกำหนดรูปร่างได้
- public void setSize(int w, int h) – ใช้กำหนดขนาดของ JFrame โดย w คือ ความกว้าง h คือ ความสูง หน่วยเป็น pixel
- public void setTitle(String s) – ใช้กำหนดชื่อบน Title Bar
- public void setVisible(boolean b) – ใช้กำหนดค่าการแสดงผล JFrame
- public void setDefaultCloseOperation(int operation) – ใช้กำหนดว่าเมื่อ กดปุ่ม แล้วให้โปรแกรมเป็นอย่างไร

22

Panel

- Container
 - Top Level
 - Non - top Level ←
- Simple Component
- Text Component

23

Panel

- สร้างจาก Class JPanel

รูปแบบ

```
JPanel pName = new JPanel();
```

```
JPanel pName = new JPanel(manager);
```

โดยที่ pName เป็นชื่อ Object ที่ประกาศจาก Class JPanel
manager เป็นรูปแบบของ LayoutManager ที่ใช้สำหรับจัดวาง คอมโพเนนต์บน Panel

Default Layout Manager จะเป็นแบบ Border Layout

24

Panel

- You can use new JPanel() to create a panel with a default FlowLayoutManager
- or new JPanel(LayoutManager) to create a panel with the specified layout manager.
- Use the add(Component) method to add a component to the panel.

For example

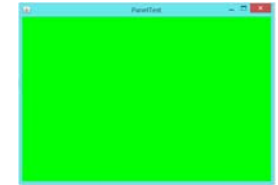
```
JPanel p = new JPanel();
```

Default JPanel Layout Manager จะเป็นแบบ Flow Layout

25

Panel

```
3 import java.awt.Color;
4 import javax.swing.JFrame;
5 import javax.swing.JPanel;
6
7 public class Lec02_FrameTest
8 {
9     public static void main(String[] args)
10    {
11        JFrame frame = new JFrame("PanelTest");
12        frame.setSize(700, 500);
13        frame.setVisible(true);
14
15        JPanel panel = new JPanel();
16        panel.setBackground(Color.green);
17
18        frame.add(panel);
19    }
20 }
```



26

The Color Class

- You can set colors for GUI components by using the java.awt.Color class. Colors are made of red, green, and blue components, each of which is represented by a byte value that describes its intensity, ranging from 0 (darkest shade) to 255 (lightest shade).

รูปแบบ

```
Color c = new Color( r , g , b );
```

Example

```
Color c = new Color(228,100,255);
```



27

The Color Class

- java.awt.Color

Thirteen standard colors (black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow)

Since JDK 1.4, you can also use the new constants:

BLACK, BLUE, CYAN, DARK_GRAY, GRAY, GREEN, LIGHT_GRAY, MAGENTA, ORANGE, PINK, RED, WHITE, and YELLOW.

28

Button

- Container
 - Top Level
 - Non - top Level
- Simple Component ←
- Text Component

29

Button

เป็นปุ่มใช้งานทั่วไปที่สร้างจากคลาส JButton

รูปแบบ

```
JButton buttonName = new JButton(text);
```

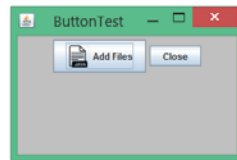
```
JButton buttonName = new JButton(text, icon);
```

โดยที่	buttonName	เป็นชื่อ Object ที่ประกาศจาก Class JButton
	text	เป็นข้อความที่ต้องการให้แสดงบนปุ่ม
	icon	เป็นรูปที่ต้องการให้แสดงบนปุ่ม

30

Button

```
3=import java.awt.Color;
4 import javax.swing.Icon;
5 import javax.swing.ImageIcon;
6 import javax.swing.JButton;
7 import javax.swing.JFrame;
8 import javax.swing.JPanel;
9
10 public class Lec03_ButtonTest
11 {
12     public static void main(String[] args)
13     {
14         JFrame frame = new JFrame("ButtonTest");
15         JPanel panel = new JPanel();
16         panel.setBackground(Color.LIGHT_GRAY);
17
18         String str1 = "E:/CPE/Lecture/Year2016/CODE/JAVA/w08_workspace_java/Lec8_GUI/src/P81/java_icon01.png";
19         Icon icon = new ImageIcon(str1);
20         JButton b1 = new JButton("Add Files", icon);
21         JButton b2 = new JButton("Close");
22         panel.add(b1);
23         panel.add(b2);
24         frame.add(panel);
25
26         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
27         frame.setSize(300, 200);
28         frame.setVisible(true);
29     }
30 }
31 }
```



31

Image Icons

- Java uses the javax.swing.ImageIcon class to represent an icon.
- An icon is a **fixed-size picture**; typically it is small and used to decorate components.
- Images are normally stored in image files.
- You can use **new ImageIcon(filename)** to construct an image icon.

```
ImageIcon icon = new ImageIcon("image/us.gif");
setLayout(new GridLayout(1, 4, 5, 5));
add(new JLabel(icon));
```

32

Label

- Container
 - Top Level
 - Non - top Level
- Simple Component ←
- Text Component

33

Label

ใช้แสดงข้อความแบบบรรทัดเดียว สร้างจากคลาส JLabel

รูปแบบ

```
JLabel lName = new JLabel(text, icon, Alignment);
```

โดยที่	lName	เป็นชื่อ Object ที่ประกาศจาก Class JLabel
	text	เป็นข้อความที่ต้องการให้แสดงบน Label
	icon	เป็นรูปที่ต้องการให้แสดงบน Label
	Alignment	เป็นตำแหน่งการจัดวางตามแนวนอนบน Label
		SwingConstants.LEFT จัดวางแบบชิดซ้าย
		SwingConstants.CENTER จัดวางแบบกึ่งกลาง
		SwingConstants.RIGHT จัดวางแบบชิดขวา

34

Label

ใช้สร้างวัตถุข้อความ(Label)

ใช้คำสั่งต่อไปนี้

```
JLabel<ชื่อวัตถุ> = new JLabel(text);  
JLabel<ชื่อวัตถุ>= new JLabel(text, alignment);  
JLabel<ชื่อวัตถุ>= new JLabel(image);  
JLabel<ชื่อวัตถุ>= new JLabel(image, alignment);  
JLabel<ชื่อวัตถุ>= new JLabel(text, image, alignment);
```

โดยค่าต่างๆ คือ

text - ข้อความที่ต้องการแสดง
image - ที่อยู่ของไฟล์รูปภาพที่จะใช้แสดง
alignment - การจัดเรียงใน Container // SwingConstants.(LEFT, Center, RIGHT)

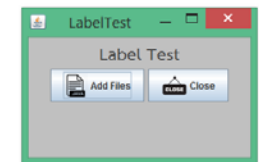
ตัวอย่างเช่น

```
JLabel label1 = new JLabel( "Name :", "C:/bamboo.gif",  
SwingConstants.RIGHT);
```

35

Label

```
18* public static void main(String[] args)  
19 {  
20     Font font1 = new Font("TH SarabunPSK", Font.BOLD, 30);  
21  
22     JFrame frame = new JFrame("LabelTest");  
23     JPanel panel = new JPanel();  
24     panel.setBackground(Color.LIGHT_GRAY);  
25  
26     JLabel label1 = new JLabel("Label Test", SwingConstants.CENTER);  
27     label1.setFont(font1);  
28     label1.setPreferredSize(new Dimension(200, 30));  
29     label1.setForeground(Color.DARK_GRAY);  
30  
31     String str1 = "E:/CPE/Lecture/Year2016/CODE/JAVA/w08_workspace_java/Lec8_GUI/src/P81/java_icon01.png";  
32     Icon icon1 = new ImageIcon(str1);  
33  
34     String str2 = "E:/CPE/Lecture/Year2016/CODE/JAVA/w08_workspace_java/Lec8_GUI/src/P81/close_icon01.png";  
35     Icon icon2 = new ImageIcon(str2);  
36  
37     JButton b1 = new JButton("Add Files",icon1);  
38     JButton b2 = new JButton("Close",icon2);  
39  
40     panel.add(label1);  
41     panel.add(b1);  
42     panel.add(b2);  
43     frame.add(panel);  
44  
45     frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
46     frame.setSize(300, 200);  
47     frame.setVisible(true);  
48  
49 }  
50 }
```



36

The Font Class

Font Names

Standard font names that are supported in all platforms are:
SansSerif, Serif,
Monospaced, Dialog, or
DialogInput.

Font Style

Font.PLAIN (0),
Font.BOLD (1),
Font.ITALIC (2),
and Font.BOLD + Font.ITALIC (3)

```
Font myFont = new Font(name, style, size);
```

Example:

```
Font myFont1 = new Font("SansSerif ", Font.BOLD+Font.ITALIC, 20);  
Font myFont2 = new Font("Serif", Font.BOLD, 12);
```

```
JButton jbtcancel = new JButton("Cancel");  
jbtcancel.setFont(myFont1);
```



37

The Font Class

Finding All Available Font Names

```
GraphicsEnvironment e = GraphicsEnvironment.getLocalGraphicsEnvironment();  
String[] fontnames = e.getAvailableFontFamilyNames();
```

```
for (int i = 0; i < fontnames.length; i++)  
    System.out.println(fontnames[i]);
```

```
Swis721 WGL4 BT  
Sylfaen  
Symbol  
Tahoma  
TeamViewer11  
Tempus Sans ITC  
TH Sarabun New  
TH SarabunPSK  
Times New Roman  
Traditional Arabic  
Trebuchet MS  
Tunga
```

Show All Fonts

38

Setting Colors

➤ You can use the following methods to set the component's background and foreground colors:

➤ `setBackground(Color c)`

➤ `setForeground(Color c)`



➤ Example:

➤ `jbt.setBackground(Color.YELLOW);`

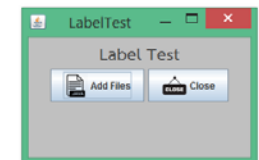
➤ `jbt.setForeground(Color.RED);`

```
jbt.setForeground(Color.red);
```

39

Label

```
18 public static void main(String[] args)  
19 {  
20     Font font1 = new Font("TH SarabunPSK", Font.BOLD, 30);  
21  
22     JFrame frame = new JFrame("LabelTest");  
23     JPanel panel = new JPanel();  
24     panel.setBackground(Color.LIGHT_GRAY);  
25  
26     JLabel label1 = new JLabel("Label Test", SwingConstants.CENTER);  
27     label1.setFont(font1);  
28     label1.setPreferredSize(new Dimension(200, 30));  
29     label1.setForeground(Color.DARK_GRAY);  
30  
31     String str1 = "E:/CPE/Lecture/Year2016/CODE/JAVA/w08_workspace_java/Lec8_GUI/src/P81/java_icon01.png";  
32     Icon icon1 = new ImageIcon(str1);  
33  
34     String str2 = "E:/CPE/Lecture/Year2016/CODE/JAVA/w08_workspace_java/Lec8_GUI/src/P81/close_icon01.png";  
35     Icon icon2 = new ImageIcon(str2);  
36  
37     JButton b1 = new JButton("Add Files", icon1);  
38     JButton b2 = new JButton("Close", icon2);  
39  
40     panel.add(label1);  
41     panel.add(b1);  
42     panel.add(b2);  
43     frame.add(panel);  
44  
45     frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
46     frame.setSize(300, 200);  
47     frame.setVisible(true);  
48  
49 }  
50 }
```



40

Text Field and Password Field

- Container
 - Top Level
 - Non - top Level
- Simple Component
- Text Component ←

41

Text Field

เป็นคอมโพเนนต์สำหรับรับข้อความจากผู้ใช้งาน สร้าง JTextField

รูปแบบ

```
JTextField textName = new JTextField(text, Length);
```

โดยที่	textName	เป็นชื่อออบเจกต์ที่ประกาศจากคลาส JTextField
	text	เป็นข้อความที่ต้องการให้แสดงบน Text Field
	Length	เป็นจำนวนตัวอักษรจะแสดงใน Text Field

42

Text Field

ใช้สร้างวัตถุกล่องข้อความ (Text area)

ใช้คำสั่งต่อไปนี้

```
JTextArea<ชื่อObject> = new JTextArea();  
JTextArea<ชื่อObject> = new JTextArea( row, column );  
JTextArea<ชื่อObject> = new JTextArea( text,row, column);  
JTextArea<ชื่อObject> = new JTextArea(); <ชื่อObject>.setText(text);
```

โดยค่าต่างๆ คือ

row -จำนวนแถวที่จะรับค่า
column -จำนวนตัวอักษรที่จะรับค่า
text -ข้อความที่ต้องการแสดงเป็นค่าเริ่มต้น

Ex JTextAreatextArea1 = new JTextArea(); textArea1.setText ("Computer");

43



Password Field

เป็นคอมโพเนนต์สำหรับรับข้อมูลรหัสผ่านโดยไม่แสดงค่าข้อมูล

สร้างจาก Class JPasswordField

รูปแบบ

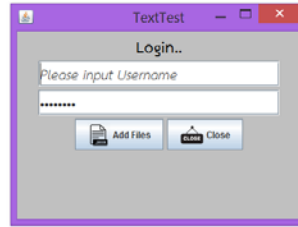
```
JPasswordField pw = new JPasswordField(text, Length);
```

โดยที่	pw	เป็นชื่อออบเจกต์ที่ประกาศจากคลาส JPasswordField
	text	เป็นรหัสผ่านที่ต้องการแสดงบน Password Field
	Length	เป็นจำนวนตัวอักษรของรหัสผ่าน

44

Text + Password Field

```
19 public static void main(String[] args)
20 {
21     Font font1 = new Font("TH SarabunPSK", Font.BOLD, 30);
22     Font font2 = new Font("TH SarabunPSK", Font.ITALIC, 26);
23
24     JFrame frame = new JFrame("TextTest");
25     JPanel panel = new JPanel();
26     panel.setBackground(Color.LIGHT_GRAY);
27
28     JLabel label1 = new JLabel("Login..", SwingConstants.CENTER);
29     label1.setFont(font1);
30     label1.setPreferredSize(new Dimension(200, 30));
31     label1.setForeground(Color.BLACK);
32
33     JTextField text1 = new JTextField("Please input Username", 20);
34     JPasswordField pass1 = new JPasswordField("Password", 20);
35     text1.setFont(font2);
36     pass1.setFont(font2);
37
38     String str1 = "E:/CPE/Lecture/Year2016/CODE/JAVA/w08_workspace_java/Lec8_GUI/src/P81/java_icon01.png";
39     Icon icon1 = new ImageIcon(str1);
40
41     String str2 = "E:/CPE/Lecture/Year2016/CODE/JAVA/w08_workspace_java/Lec8_GUI/src/P81/close_icon01.png";
42     Icon icon2 = new ImageIcon(str2);
43
44     JButton b1 = new JButton("Add Files", icon1);
45     JButton b2 = new JButton("Close", icon2);
46
47     panel.add(label1);
48     panel.add(text1);
49     panel.add(pass1);
50     panel.add(b1);
51     panel.add(b2);
52     frame.add(panel);
53
54     frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
55     frame.setSize(400, 300);
56     frame.setVisible(true);
57 }
```



45

RadioButton

- เป็นคอมโพเนนต์สำหรับการสร้างตัวเลือกที่สามารถเลือกได้เพียงหนึ่งตัวเลือก
- สร้างจากคลาส JRadioButton และใช้คลาส ButtonGroup ในการจัดกลุ่มตัวเลือกต่างๆ ให้อยู่ในกลุ่มเดียวกัน

รูปแบบ `JRadioButton radioName = new JRadioButton(text, icon, boolValue);`

โดยที่

radioName	เป็นชื่อออบเจกต์ที่ประกาศจากคลาส JRadioButton
text	เป็นข้อความที่ต้องการให้แสดงบน RadioButton
icon	เป็นไอคอนที่ต้องการให้แสดงบน RadioButton
boolValue	เป็นสถานการณ์เลือก ถ้ากำหนดเป็น true คือ ถูกเลือก

Class ButtonGroup มีรูปแบบการใช้งาน ดังนี้

```
public ButtonGroup();
มีเมธอด add() และ remove() สำหรับการเพิ่มหรือลบตัวเลือก
```

46

List

- เป็นคอมโพเนนต์สำหรับสร้างรายการข้อมูลที่สามารถเลื่อนรายการขึ้นลง
- สร้างจากคลาส JList และใช้คลาส JScrollPane จัดการรายการข้อมูลใน List

รูปแบบ

```
JList listName = new JList();
JList listName = new JList(dataModel);
JList listName = new JList(items);
```

โดยที่

listName	เป็นชื่อออบเจกต์ที่ประกาศจากคลาส Jlist
dataModel	เป็นการกำหนดให้เลือกรายการได้ 1 หรือมากกว่า 1 รายการ
items	เป็นข้อมูลประเภทของรายการ Object หรือ Vector

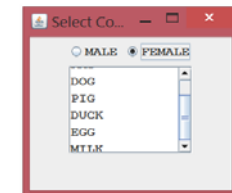
47

Radio + List

```
public static void main(String[] args)
{
    JFrame f;
    JPanel p;
    ButtonGroup bg;
    JRadioButton n1, n2;
    JList AList;

    String AniList[] = {"CAT", "RAT", "DOG", "PIG", "DUCK", "EGG", "MILK"};
    Font fn = new Font("Courier New", Font.BOLD, 16);
    f = new JFrame("Select Component Test");
    p = new JPanel();
    bg = new ButtonGroup();
    n1 = new JRadioButton("MALE", true);
    n2 = new JRadioButton("FEMALE");
    bg.add(n1);
    bg.add(n2);
    n1.setFont(fn);
    n2.setFont(fn);

    AList = new JList(AniList);
    AList.setFont(fn);
    AList.setPreferredSize(new Dimension(135, 140));
    AList.setVisibleRowCount(5);
    AList.setSelectionMode(ListSelectionModel.MULTIPLE_INTERVAL_SELECTION);
    JScrollPane panelist = new JScrollPane(AList);
    p.add(n1);
    p.add(n2);
    p.add(panelist);
    f.add(p);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    f.setSize(230, 200);
    f.setVisible(true);
}
```



48

ScrollPane

- เป็นคอมโพเนนต์สำหรับสร้าง Scroll bar
- สร้างจากคลาส JScrollPane

รูปแบบ

```
JScrollPane panelist = new JScrollPane(listName);
```

โดยที่	panelist	เป็นชื่อออบเจ็กต์ที่ประกาศจากคลาส JScrollPane
	listName	เป็นรายการข้อมูลประเภท List

49

CheckBox

- เป็นคอมโพเนนต์สำหรับการสร้างตัวเลือกที่เลือกได้มากกว่าหนึ่งตัวเลือก
- สร้างจากคลาส JCheckBox มีสถานะเป็น true หรือ false

รูปแบบ

```
JCheckBox checkName = new JCheckBox(text, icon);
```

โดยที่	checkName	เป็นชื่อออบเจ็กต์ที่ประกาศจากคลาส JCheckBox
	text	เป็นข้อความที่ต้องการให้แสดงบน CheckBox
	icon	เป็นไอคอนที่ต้องการให้แสดงบน CheckBox

50

ComboBox

- เป็นคอมโพเนนต์สำหรับแสดงรายการข้อมูล ซึ่งแสดงได้เพียง 1 รายการและ เลือกข้อมูลในได้เพียง 1 รายการเท่านั้น
- สร้างจากคลาส JComboBox มีรูปแบบการใช้งาน ดังนี้

รูปแบบ

```
JComboBox comboName = new JComboBox(items);
```

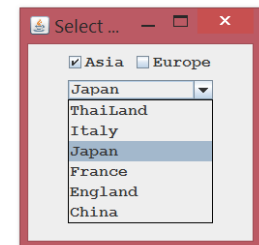
โดยที่	comboName	เป็นชื่อออบเจ็กต์ที่ประกาศจากคลาส JComboBox
	items	เป็นรายการข้อมูลประเภท Object หรือ Vector

51

Scroll + CheckBox + Combo

```
public static void main(String[] args)
{
    JFrame f; JPanel p; JCheckBox myCheckbox1, myCheckbox2;
    JComboBox combo;
    Font fn = new Font("Courier New", Font.BOLD, 16);
    String country[] = {"Thailand", "Italy", "Japan", "France", "England", "China"};
    f = new JFrame("Select Component Test");
    p = new JPanel();
    myCheckbox1 = new JCheckBox("Asia");
    myCheckbox2 = new JCheckBox("Europe");
    myCheckbox1.setFont(fn);
    myCheckbox2.setFont(fn);

    combo = new JComboBox(country);
    combo.setFont(fn);
    combo.setPreferredSize(new Dimension(150, 20));
    combo.setSelectedItem("Japan");
    combo.setMaximumRowCount(6);
    p.add(myCheckbox1);
    p.add(myCheckbox2);
    p.add(combo);
    f.add(p);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    f.setSize(250, 250);
    f.setVisible(true);
}
```



52

Menu

เป็นเครื่องมือให้ผู้ใช้งานเลือกใช้คำสั่งต่างๆ จากรายการซึ่งถูกสร้างขึ้นจากเมนูบาร์ มีขั้นตอนดังนี้ กำหนดเมนูบาร์ให้กับ Frame โดยใช้เมธอด setJMenuBar

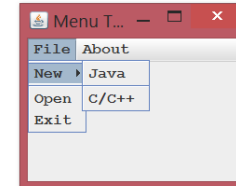
```
frameName.setJMenuBar(menuubar);
```

โดยที่	menuubar	เป็นชื่อเมนูบาร์
	menuName	เป็นชื่อรายการเมนูในเมนู
	itemName	เป็นข้อความที่แสดงในรายการเมนู
	menuItemName	เป็นชื่อรายการเมนูย่อยในเมนู
	subItemName	เป็นข้อความที่แสดงในรายการเมนูย่อย

53

Scroll + CheckBox + Combo

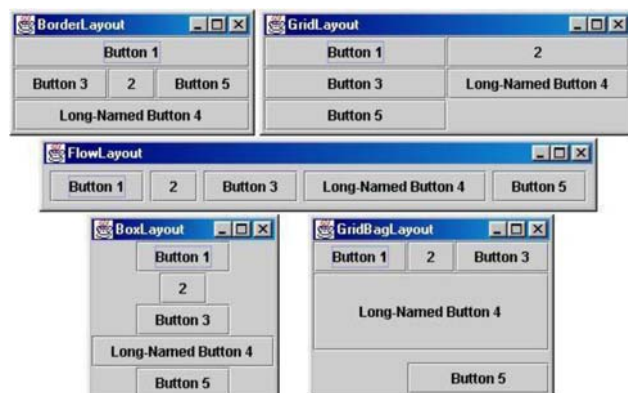
```
public static void main(String[] args)
{
    Font fn = new Font("Courier New", Font.BOLD, 16);
    JFrame f;
    JMenuBar menuubar;
    JMenu menuFile, menuNew, menuAbout;
    JMenuItem menuN1, menuN2, menuOpen, menuExit;
    f = new JFrame("Menu Test"); menuubar = new JMenuBar();
    menuFile = new JMenu("File"); menuAbout = new JMenu("About");
    menuNew = new JMenu("New");
    menuN1 = new JMenuItem("Java"); menuN2 = new JMenuItem("C/C++");
    menuNew.add(menuN1); menuNew.addSeparator(); menuNew.add(menuN2);
    menuOpen = new JMenuItem("Open");
    menuExit = new JMenuItem("Exit");
    menuFile.setFont(fn);
    menuNew.setFont(fn);
    menuN1.setFont(fn);
    menuN2.setFont(fn);
    menuOpen.setFont(fn);
    menuExit.setFont(fn);
    menuAbout.setFont(fn);
    menuFile.add(menuNew);
    menuFile.addSeparator();
    menuFile.add(menuOpen);
    menuFile.add(menuExit);
    menuubar.add(menuFile);
    menuubar.add(menuAbout);
    f.setJMenuBar(menuubar);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    f.setSize(250, 200);
    f.setVisible(true);
}
```



54

Layout Managers

Layout Managers ใช้สำหรับจัดแสดง component ออกทางจอภาพ



Default layout ของ JFrame: FlowLayout

55

Layout Managers

- FlowLayout
- GridLayout
- BorderLayout
- Several other layout managers

56

Flow Layout

- Write a program that adds three labels and text fields into the content pane of a frame with a FlowLayout manager.
- The components are arranged in the container from left to right in the order which they were added



57

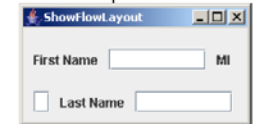
Flow Layout

```
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.JFrame;
import java.awt.FlowLayout;

public class ShowFlowLayout extends JFrame {
    public ShowFlowLayout() {
        // Set FlowLayout, aligned left with horizontal gap 10
        // and vertical gap 20 between components
        setLayout(new FlowLayout());

        // Add labels and text fields to the frame
        add(new JLabel("First Name"));
        add(new JTextField(8));
        add(new JLabel("MI"));
        add(new JTextField(1));
        add(new JLabel("Last Name"));
        add(new JTextField(8));
    }

    /** Main method */
    public static void main(String[] args) {
        ShowFlowLayout frame = new ShowFlowLayout();
        frame.setTitle("ShowFlowLayout");
        frame.setLocationRelativeTo(null); // New since JDK 1.4
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(200, 200);
        frame.setVisible(true);
    }
}
```



Grid Layout

- Rewrite the program in the preceding example using a GridLayout manager instead of a FlowLayout manager to display the labels and text fields.



```
setLayout(new GridLayout(3, 2));
```

59

Border Layout

- The BorderLayout manager divides the container into five areas: East, South, West, North, and Center.
- Components are added to a BorderLayout by using the add method.

```
add(Component, constraint),
where constraint is
BorderLayout.EAST,
BorderLayout.SOUTH,
BorderLayout.WEST,
BorderLayout.NORTH, or
BorderLayout.CENTER.
```

60

Border Layout



```
add(Component, constraint),  
where constraint is  
BorderLayout.EAST,  
BorderLayout.SOUTH,  
BorderLayout.WEST,  
BorderLayout.NORTH, or  
BorderLayout.CENTER.
```

North, South ยึดตามแนวนอน
West, East ยึดตามแนวตั้ง
Center ยึดตามแนวนอนและแนวตั้ง

61

Event

Programming Fundamentals II

62

Event

- เป็นการจัดการในส่วนโต้ตอบระหว่างผู้ใช้ กับ GUI ด้วย เช่น การคลิกที่ปุ่ม Button หรือการกด Enter บนแป้นพิมพ์ที่ Text Field เป็นต้น
- จะต้องตรวจสอบเหตุการณ์ก่อนว่าผู้ใช้มีการกระทำกับคอมโพเนนต์ที่เหตุการณ์ใดบ้าง ซึ่งเป็นหน้าที่ของออบเจกต์ที่เป็น Event Listener
- จากนั้นนำ Event Listener ไปผูกติดกับคอมโพเนนต์ เช่น ต้องการตรวจสอบเหตุการณ์ที่ปุ่ม CloseButton โดยสร้างออบเจกต์จากคลาส ButtonListener เพื่อให้ออบเจกต์ดังกล่าวทำหน้าที่เป็น Event Listener และนำไปผูกติดกับปุ่ม CloseButton

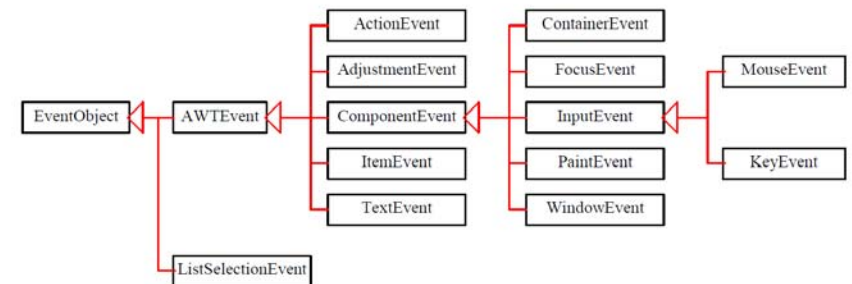
รูปแบบ

```
buttonName.addActionListener(new ButtonListener());
```

โดยที่ buttonName เป็นชื่อออบเจกต์ที่ประกาศจากคลาส JButton
 ButtonListener เป็นออบเจกต์จากคลาส ButtonListener

63

Event Class



The root class of the event classes is java.util.EventObject.

The hierarchical relationships of some event classes are shown above.

64

Event User Actions

User Action	Source Object	Event Type Generated
Click a button	JButton	ActionEvent
Press return on a text field	JTextField	ActionEvent
Select a new item	JComboBox	ItemEvent, ActionEvent
Select item(s)	JList	ListSelectionEvent
Click a check box	JCheckBox	ItemEvent, ActionEvent
Click a radio button	JRadioButton	ItemEvent, ActionEvent
Select a menu item	JMenuItem	ActionEvent
Move the scroll bar	JScrollBar	AdjustmentEvent
Window opened, closed, etc.	Window	WindowEvent
Mouse pressed, released, etc.	Component	MouseEvent
Key released, pressed, etc.	Component	KeyEvent
Component added or removed etc.	Container	ContainerEvent
Component moved, etc.	Component	ComponentEvent
Component gained or lost focus	Component	FocusEvent

☞ Table lists external user actions, source objects, and event types generated.

65

Event Method

ประเภทเหตุการณ์	อินเทอร์เฟส	เมธอด
ActionEvent	ActionListener	actionPerformed(ActionEvent)
AdjustmentEvent	AdjustmentListener	adjustmentValueChanged(AdjustmentEvent)
ComponentEvent	ComponentListener	componentMoved(ComponentEvent) componentHidden(ComponentEvent) componentResized(ComponentEvent) componentShown(ComponentEvent)
ContainerEvent	ContainerListener	componentAdded(Container Event) componentRemoved(Container Event)
FocusEvent	FocusListener	focusGained(FocusEvent) focusLost(FocusEvent)
WindowEvent	WindowListener	windowClosing(WindowEvent) windowOpened(WindowEvent) windowIconified(WindowEvent) windowDeiconified(WindowEvent) windowClosed(WindowEvent) windowActivated(WindowEvent) windowDeactivated(WindowEvent)

66

Inner Class

- การเขียนโปรแกรมในส่วนจัดการเหตุการณ์ สามารถจัดการให้อยู่ภายใน Inner Class
- โดยสร้างเป็นคลาสที่มีการ implements อินเทอร์เฟสซึ่งสอดคล้องกับเหตุการณ์ที่เกิดขึ้น และ
- กำหนดหน้าที่การทำงานให้กับเมธอดในอินเทอร์เฟสนั้นๆ

รูปแบบ

```
private class ButtonListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        if(e.getSource() == buttonName) {
            // ...
        }
    }
}
```

โดยที่ ButtonListener เป็นออบเจกต์จากคลาส ButtonListener
 ActionListener เป็นเหตุการณ์ที่ต้องการตรวจจับ
 actionPerformed เป็นชื่อเมธอด
 buttonName เป็นชื่อออบเจกต์ที่ประกาศจากคลาส JButton

67

Close Button

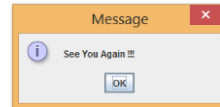
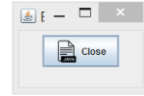
```
3=import java.awt.*;
4 import java.awt.event.*;
5 import javax.swing.*;
6 public class Lec09_CloseEvent extends JFrame
7 {
8     private JPanel p; Icon ani; JButton b;
9     public Lec09_CloseEvent(String title)
10    {
11        super(title);
12        p = new JPanel();
13        String str1 = "E:/CPE/Lecture/Year2016/CODE/JAVA/w08_workspace_java/Lec8_GUI/src/P81/java_icon01.png";
14        ani = new ImageIcon(str1);
15        b = new JButton("Close", ani);
16        b.addActionListener(new ButtonListener());
17        p.add(b);
18        add(p);
19    }
20    private class ButtonListener implements ActionListener
21    {
22        public void actionPerformed(ActionEvent e)
23        {
24            if (e.getSource() == b)
25            {
26                JOptionPane.showMessageDialog(null, "See You Again !!!");
27                System.exit(0);
28            }
29        }
30    }
31    public static void main(String[] args)
32    {
33        Lec09_CloseEvent b = new Lec09_CloseEvent("Button Test");
34        b.setSize(170, 120);
35        b.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
36        b.setVisible(true);
37    }
38 }
39 }
```



68

Close Button

```
3import java.awt.*;
4import java.awt.event.*;
5import javax.swing.*;
6public class Lec09_CloseEvent extends JFrame
7{
8    private JPanel p; Icon ani; JButton b;
9    public Lec09_CloseEvent(String title)
10    {
11        super(title);
12        p = new JPanel();
13        String str1 = "E:/CPE/Lecture/Year2016/CODE/JAVA/w08_workspace_java/Lec8_GUI/src/P81/java_icon01.png";
14        ani = new ImageIcon(str1);
15        b = new JButton("Close", ani);
16        b.addActionListener(new ButtonListener());
17        p.add(b);
18        add(p);
19    }
20    private class ButtonListener implements ActionListener
21    {
22        public void actionPerformed(ActionEvent e)
23        {
24            if (e.getSource() == b)
25            {
26                JOptionPane.showMessageDialog(null, "See You Again !!!");
27                System.exit(0);
28            }
29        }
30    }
31    public static void main(String[] args)
32    {
33        Lec09_CloseEvent b = new Lec09_CloseEvent("Button Test");
34        b.setSize(170, 120);
35        b.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
36        b.setVisible(true);
37    }
38 }
39 }
```



69

END

Full Event + UI Next week

70