

•Chapter 7

#### Event-Driven Programming Model





## By Dr. Paween Khoenkaw Computer Science MJU



Flow of the program

Procedural programing model คือการโปรแกรมในลักษณะ ของการกำหนดการทำงานเป็นแบบ ทีละขั้นตอน (step by step) เป็นการทำงานในระดับต่ำ



Flow of the program

Event-Driven programing model คือการโปรแกรมในลักษณะ ของการกำหนดการทำงานเป็นแบบ ต้องทำอะไรบ้างเมื่อเกิด เหตุการณ์ (Event)ต่างๆขึ้น เป็นการทำงานในระดับสูง



Flow of the program

Event-Driven programing model คือการโปรแกรมในลักษณะ ของการกำหนดการทำงานเป็นแบบ ต้องทำอะไรบ้างเมื่อเกิด เหตุการณ์ (Event)ต่างๆขึ้น เป็นการทำงานในระดับสูง



Event-Driven model

Event-Driven programing model จะประกอบด้วย 2 ส่วนหลักๆคือ 1 Event Raiser สร้างเหตุการณ์ต่างๆ หรือทำหน้าที่ส่ง Event 2 Event Listener เป็น Method ที่จะถูกเรียกขึ้นมาเมื่อเกิดเหตุการณ์ต่างๆ

Raiser และ Listener เรียกรวมๆว่า Event Handling จะต้องใช้คู่กันเสมอ เช่น Mouse Click Raiser จะใช้คู่กับ Mouse Click Listener ใช้ผิดตัวไม่ได้

เราจะใส่ code ทำงานเมื่อมีการ click mouse ด้วยการ

Override ที่ Mouse Clock Listener



**Event Listener** 



**Event Raiser** 

Event-Driven model

Event-Driven programing model จะประกอบด้วย 2 ส่วนหลักๆคือ 1 Event Raiser สร้างเหตุการณ์ต่างๆ หรือทำหน้าที่ส่ง Event 2 Event Listener เป็น Method ที่จะถูกเรียกขึ้นมาเมื่อเกิดเหตุการณ์ต่างๆ



Event-Driving Programming Model

#### User Interface

marsUmarsmain 🐩 💲 pwd
/home/mars
mars@marsmain ~ \$ cd /usr/portage/app-shells/bash
mars@marsmain_/usr/portane/app-sbells/basb_\$_lsal
total 130
druvr-vr-v 2 portage portage 1824 Jul 25 18:85
di war ya 2 portage portage 1024 001 2 10.00
urwxr-xr-x 35 por tage por tage 1824 Hug 7 22:39
-ru-rr 1 root root 35000 JUI 25 18:00 ChangeLog
-rw-rr 1 root root 2/002 Jul 25 10:06 Manifest
-rw−r−−r−− 1 portage portage 4645 Mar 23 21:37 bash−3.1_p17.ebuild
−rw−r−−r−− 1 portage portage 5977 Mar 23 21:37 bash−3.2_p39.ebuild
–rw−r−−r−− 1 portage portage 6151 Apr 5 14:37 bash−3.2_p48–r1.ebuild
-rw−r−−r−− 1 portage portage 5988 Mar 23 21:37 bash−3.2_p48.ebuild
-rw-rr 1 portage portage 5643 Apr 5 14:37 bash-4.0_p10-r1.ebuild
-rw-rr 1 portage portage 6230 Apr 5 14:37 bash-4.0 p10.ebuild
-rw-rr 1 portage portage 5648 Apr 14 05:52 bash-4.0 p17-r1.ebuild
-rw-rr 1 portage portage 5532 Apr 8 10:21 bash-4.0 p17 ebuild
-rw-rr 1 portage portage 5560 May 30 03:35 bash-4 0 p24 ebuild
-ru-r 1 root root 5660 Jul 25 00:43 bash-4 0 28 ebuild
an with with a point age point age 2040 hag 30 03.35 miles
-rw-rr i pur tage pur tage 460 reb 9 04:55 metauata.xml
marsemarsmain /usr/portage/app_snells/bash % Cat metadata.xmi
<pre></pre> // version="1.0" encoding="0 F-8"?>
pkgmetadata SYSTEM "<u http://www.gentoo.org/dtd/metadata.dtd">
<pre><pre>cpkgmetadata&gt;</pre></pre>
<herd>base-system</herd>
<use></use>
<flag name="bashlogger">Log ALL commands typed into bash; should ONLY be</flag>
used in restricted environments such as honeupots
<flag name="net">Enable /dev/tcp/host/port redirection</flag>
<flag pame="plugips">8dd support for loading builting at runtime via</flag>
'enable'//flan
Zhicas
(vprgumetauata)
lar Sellar Sellar / USF / por tage/app-silerrs/basil \$ Sudo / etc/init.u/bruetootil status
Passwuru:
* status: started
mars@marsmain /usr/portage/app-shells/bash \$ ping -q -cl en.wikipedia.org
PING rr.esams.wikimedia.org (91.198.174.2) 56(84) bytes of data.
––– rr.esams.wikimedia.org ping statistics –––
1 packets transmitted, 1 received, 0% packet loss, time 2ms
rtt min/avg/max/mdev = 49,820/49,820/49,820/0,000 ms
mars@marsmain /usc/portage/app-shells/bash \$ grep -i /dev/sda /etc/fstab   cutfields=-
/dev/sda1 /boot
marsemars main /usr/portage/app-snells/bash % date
Sat Hug 6 02:42:24 NSD 2009
mars@marsmain /usr/portage/app-shells/bash \$ ismod
Module Size Used by
rndis_wlan 23424 0
rndis_host 8696 1 rndis_wlan
cdc_ether 5672 1 rndis_host
usbnet 18688 3 rndis_wlan,rndis_host,cdc_ether
parport pc 38424 Ø
folix 2388128 20
39548 1 perpert pc
irro udt 19272 0
nar senar sina tri vosi vpor (ayevapp-sina risvoasi) \$



Command-line Interfaces (CLI) ใช้<u>ตัวอักษร</u> และ <u>Keyboard</u> ในการป้อน Input ส่วนมากจะเป็น

Procedural programing model

Graphical User Interface (GUI) <u>ใช้ภาพ</u> Icon และ <u>Mouse</u> ในการป้อน Input ส่วนมากจะเป็น Event-Driven programing model

Java มีชุดของคลาสสำหรับสร้าง GUI ติดมาให้ 2 กลุ่มคือ

- 1 AWT (Abstract Windowing Toolkit) มีมาตั้งแต่ JDK 1.0 เก่า และล้าสมัย
- 2 Swing มีตั้งแต่ JDK 1.1 ใหม่กว่าและมีเครื่องมือให้ใช้งานมากกว่า

Swing เป็นส่วนต่อขยาย (สืบทอด)มาจากคลาส AWT ทำให้การทำงานในหลายส่วนจะ คล้ายกัน







AWT และ Swing จะมีเครื่องมือหลักให้ใช้อยู่ 4 ส่วนคือ

- 1 Components เป็นเครื่องมือที่ใช้ทำงานเฉพาะทางต่างๆ เช่นปุ่มกด (Button)
- 2 Containers ใช้สำหรับวาง Components
- 3 Layout Manager ใช้สำหรับเลือกรูปแบบการจัดวาง Components
- 4 Event handling ใช้สำหรับจัดการเหตุการณ์ต่างๆ เช่นตรวจจับการ click



คลาส Swing มักจะขึ้นต้นด้วยตัวอักษร J เสมอ



#### ส่วนประกอบแต่ละชั้นของ GUI ที่สร้างด้วย Swing



#### ส่วนประกอบแต่ละชั้นของ GUI ที่สร้างด้วย Swing



#### การดักจับ Event ต่างๆใน Swing ตัวอย่างเช่นการ Click ที่ Button



public class HelloSwing {

```
private JFrame frame;
public static void main(String[] args) {
EventQueue.invokeLater(new Runnable() {
public void run() {
try {
HelloSwing window = new HelloSwing();
window.frame.setVisible(true):
} catch (Exception e) {
e.printStackTrace();
});
public HelloSwing() {
            initialize();
}
```

```
private void initialize() {
frame = new JFrame();
frame.getContentPane().addMouseListener(new MouseAdapter() {
frame.setBounds(100, 100, 450, 300);
frame.setDefaultCloseOperation(JFrame.EXIT ON CLOSE);
frame.getContentPane().setLayout(null);
JButton btnNewButton = new JButton("Hello World");
btnNewButton.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent arg0) {
}
}):
btnNewButton.setBounds(129, 96, 157, 23);
frame.getContentPane().add(btnNewButton);
}
```

<u></u>		-	×
	Hello World		
			14

Eclipse มีเครื่องมือช่วยในการสร้าง GUI ให้ ช่วยให้เราไม่ต้องเขียน code ในการวาง Containers, Layout และ Component

แต่ต้องดาวน์โหลดมาติดตั้งเพิ่มเอง จาก <u>https://eclipse.org/windowbuilder/download.php</u> และเลือกดาวน์โหลดให้ตรงกับ version ของ Eclipse

dpplication.ja	iva 83															
Struct	ure		ାଇଅଜ	et 🗈 🖄 🗙			(default) 🔻	← >	→ +>	11:	цe		Ţ			
🍖 Components		•		0 440 445 **						-		<u> </u>			4	_
🔺 🛅 shell - "A	Application"		A Palette		1	- 6	default)	-		2			0 0	,	•	
⊳ 🖾 name	eComposite		b system	FT1 + +	1	-	de									-
Addr	ess		R Selection	L., Marquee			en US								- 0	23
a 🖃 Phon	ie		Choose co	Tab Order												
🥲 h	ome - "Home"		🗁 Composites			-	10	_								
Į́— h	omePh - " "		Composite	🖌 Canvas			Last Name									
ia o	ffice - "Office"		III Table	Tree 1												
ti o	fficePh		Diff ListRey	THE List View			Street									
🥲 n	nobile - "Mobile"		EN LISEDUX	En ciscview			City									_
<u>р</u> и	nobilePh		SortedList				0.0					7. [				_
👂 🖃 emai	IComposite		🗁 Layouts				State					2lp				
			Absolute la	E FillLayout	1		Home						Office			_
Properties	°⊙ & ≟₀ :	<b>静</b> - 昭	# GridLayout	F FormLayout	-		8									
Variable	home	*	# RowLayout		2		Mobile									
Constructor	(Constructor prop	er	Controls				Email									
🗏 Style	0	_	Push Button	Check Box	1											
border	false		C Paris Date	- Check box												
separator	talse		Kadio Button	Cabel												
wrap			t Text	Combo												
chadow	SHADOW OUT	_	List	💷 ProgressBar												
alion	LEFT		10 Slider	Browser												
	(org.eclipse.swt.la	vo	Table	E TableColu		1										
Class	org.eclipse.swt.wid	dg	m Table	B Tablecold												
alignment	LEFT		# Tablettem	eg Tree												
background	240,240,240	-	t, Treeltern													
enabled	✓ true		🗁 Menu													
font	MS Shell Dig 9		🔤 Menu Bar	🚚 Popup Menu												
foreground	0,0,0	-	E Cascade M	A Menultem												
image			B Radio Men	Check Men												
text	Home		• Raulo Men	Check Men									W	indowB	uilder Pro	
toolTipText			Separator										W	ww.win	lowbuilde	spro.com
🗲 Source 📑 I	Design															

#### เลือกติดตั้งให้ครบทั้ง Swing , SWT และ Windows Builder

#### 🖨 Install

#### Install Details

A Your original request has been modified. See the details.

Name	Version	ld
🚯 Swing Designer	1.9.0	org.eclipse.wb.swing.feature.featu
Swing Designer Documentation	1.9.0	org.eclipse.wb.swing.doc.user.feat
> 🖗 SWT Designer	1.9.0	org.eclipse.wb.rcp.feature.feature
> 🚯 SWT Designer Core	1.9.0	org.eclipse.wb.swt.feature.feature
🖗 SWT Designer Documentation	1.9.0	org.eclipse.wb.rcp.doc.user.featur
> 🏟 SWT Designer SWT_AWT Support	1.9.0	org.eclipse.wb.rcp.SWT_AWT_sup
> Image: SWT Designer XWT Support (requires Eclipse WTP/WST)	1.9.0	org.eclipse.wb.xwt.feature.feature
🚯 WindowBuilder Core	1.9.0	org.eclipse.wb.core.feature.feature
🖚 WindowBuilder Core Documentation	1.9.0	org.eclipse.wb.doc.user.feature.fea
🖚 WindowBuilder Core Ul	1.9.0	org.eclipse.wb.core.ui.feature.feat
🚯 WindowBuilder CSS Support	1.9.0	org.eclipse.wb.css.feature.feature
🚯 WindowBuilder GroupLayout Support	1.9.0	org.eclipse.wb.layout.group.featur
🖚 WindowBuilder Java Core	1.9.0	org.eclipse.wb.core.java.feature.fe
WindowBuilder XML Core (requires Eclipse WTP/WST)	1.9.0	org.eclipse.wb.core.xml.feature.fea

# Size: Unknown Details Your original request has been modified. "WindowBuilder Core" will be ignored because it is already installed. "WindowBuilder XML Core (requires Eclipse WTP/WST)" will be ignored because it is already installed. Volume of the image of the ima



#### Swing Framework สามารถแยกเรียนได้เป็นอีก 1 วิชา

#### Hello Swing



## Hello Swing





#### Hello Swing





<<Interface>> MouseListener +mouseClicked(MouseEvent e) +mouseEntered(MouseEvent e) +mouseExited(MouseEvent e) +mousePressed(MouseEvent e) +mouseReleased(MouseEvent e) public class Hello\_Swing {

```
public static void main(String[] args) {
  JFrame F=new JFrame();
      F.setBounds(1, 1, 400, 200);
```

```
F.<u>show(true);</u>
```

MouseListener mouse=new MouseListener(){

```
@Override
```

```
public void mouseClicked(MouseEvent arg0) {
  System.out.println("Click");
  }
```

```
@Override
public void mouseEntered(MouseEvent arg0) {
  System.out.println("Enter");
```

```
@Override
public void mouseExited(MouseEvent arg0) {
  System.out.printLn("Exit");
}
```

```
@Override
public void mousePressed(MouseEvent arg0) {
  System.out.println("Press");
}
```

```
@Override
public void mouseReleased(MouseEvent arg0) {
  System.out.println("Release");
  };
F.addMouseListener(mouse);
}
```

Hello Swing! — □ ×

มีค่าเท่ากับสร้างคลาสใหม่ที่ สืบทอดมาจาก MouseListener และนำคลาสใหม่นั้นมาสร้างเป็น instance ชื่อ mouse วิธีการนี้ใช้สำหรับสร้างหลายคลาสในไฟล์เดียว

Rroblems	@ Javadoc 😣 D
Hello_Swing [J	ava Application] C
Release	
Click	
Exit	
Enter	
Exit	
Enter	
Exit	
Enter	
Exit	

Mouse มีตั้ง 3 ปุ่ม จะรู้ได้อย่างไรว่ากำลังกดปุ่มไหนอยู่ ?

}

ถ้าต้องการจะใช้แค่ mouseClicked() จะทำอย่างไร ? ต้องเขียน code ทุก method หรือไม่ ?

MouseAdapter					
,	-	Hello Swing!	-	[	2
+mouseClicked(MouseEvent e)					
+mouseDragged(MouseEvent e)					
+mouseEntered(MouseEvent e)					
+mouseExited(MouseEvent e)					
+mouseMoved(MouseEvent e)					
+mousePressed(MouseEvent e)					
+mouseReleased(MouseEvent e)					
+mouseWheelMoved(MouseWheelEvent e)					



```
};
F.addMouseListener(mouse);
}
```

Problems	@ Javadoc	🚯 Declaration	📮 Console	8
<terminated></terminated>	Hello_Swing	[Java Application	] C:\Program	Files
Click!				
Click!				
Click!				

 $\times$ 

JFrame เป็น Container JButton เป็น Component



•java.lang.Object

- java.awt.Component
  - java.awt.Container
    - javax.swing.JComponent
      - javax.swing.AbstractButton
        - javax.swing.JButton

#### JButton ต้องวางบน Container เสมอ ตัวอย่างเช่น วาง JButton ไว้ใน JFrame

```
Hello Swing: JButton
```

```
public class Hello_Swing {
public static void main(String[] args) {
JFrame F=new JFrame();
        F.setBounds(1, 1, 400, 200);
        JButton b1=new JButton("I am Button1");
        F.add(b1);
        F.show(true);
                                                         Containers : JFrame
                                            Component : JButton
}
}
        4
                                 X
                    I am Button1
```



```
public class Hello_Swing {
```

```
public static void main(String[] args) {
    JFrame F=new JFrame();
        F.setBounds(1, 1, 400, 200);
        JButton b1=new JButton("I am Button1");
        F.add(b1);
        F.<u>show(true);</u>
        b1.addMouseListener(new MouseAdapter(){
```

#### เขียนแบบนี้ก็ได้หรอ ?

```
@Override
public void mouseClicked(MouseEvent e) {
  System.out.println("Click!");
}
```

```
});
```

```
}
```

```
}
```



ตอบ: ได้ เพราะเราไม่ต้องการนำเอา instance ไปใช้ ทำอย่างอื่น หรือต้องการเรียกใช้ mouseClick เอง แต่ต้องระวัง เพราะ MouseAdapter เป็น inner class ห้ามทำการสืบทอดต่อ

```
ลองเพิ่มปุ่มที่ 2
```

```
public class Hello Swing {
public static void main(String[] args) {
JFrame F=new JFrame();
        F.setBounds(1, 1, 400, 200);
        JButton b1=new JButton("I am Button1");
        JButton b2=new JButton("I am Button2");
        F.add(b1);
                                $
                                                         F.add(b2);
      F. show();
}
                                            I am Button2
}
```

JFrame เป็น Container ที่สามารถใส่ Component ได้แค่ตัวเดียว ต้องการใส่ Component มากกว่า 1 ต้องใช้ตัวช่วย 29

×

#### Hello Swing: JPanel

JPanel เป็น Container ที่สามารถใส่ Component ได้หลายตัว แต่ทำงานด้วยตัวเองไม่ได้ ต้องใช้ JFrame เป็นตัวช่วยในการแสดงผลบนจอ

public class Hello\_Swing {

```
public static void main(String[] args) {
  JFrame F=new JFrame();
    F.setBounds(1, 1, 400, 200);
    JButton b1=new JButton("I am Button1");
    JButton b2=new JButton("I am Button2");
    JPanel P=new JPanel();
```

```
P.add(b1);
P.add(b2);
F.setContentPane(P);
```

F.<u>show();</u>

}

}





```
Hello Swing: JPanel
public class Hello Swing {
public static void main(String[] args) {
JFrame F=new JFrame();
        F.setBounds(1, 1, 400, 200);
        JButton b1=new JButton("I am Button1");
        JButton b2=new JButton("I am Button2");
        JButton b3=new JButton("I am Button3");
        JPanel P=new JPanel();
                                                                  ×
                                      ≝
                                                              P.add(b1);
                                                  I am Button2
                                         I am Button1
                                                           I am Button3
        P.add(b2);
        P.add(b3);
        F.setContentPane(P);
        F.show();
}
```

```
ทำไมตำแหน่งปุ่มจึงเรียงจากซ้ายไปขวา ?
ต้องการเปลี่ยนตำแหน่งปุ่ม (ย้ายปุ่ม 3 ไปไว้มุมล่างขวา) จะทำได้ไหม ?
```

#### Hello Swing: JPanel

```
public class Hello_Swing {
public static void main(String[] args) {
JFrame F=new JFrame();
    F.setBounds(1, 1, 400, 200);
    JButton b1=new JButton("I am Button1");
    JButton b2=new JButton("I am Button2");
    JButton b3=new JButton("I am Button3");
    JPanel P=new JPanel();

    P.add(b1);
    P.add(b2);
    P.add(b3);
    F.setContentPane(P);
    F.<u>show();</u>
}
```

<u>م</u>	_		×
I am Button1 I am Button2	l am	Button3	

```
ทำไมตำแหน่งปุ่มจึงเรียงจากซ้ายไปขวา ?
```

ตอบ:

เพราะโดยปริยายแล้ว JPanel จะเรียง Component จากซ้ายไปขวาเสมอ ต้องการเปลี่ยนตำแหน่งปุ่ม (ย้ายปุ่ม 3 ไปไว้มุมล่างขวา) จะทำได้ไหม ? ตอบ:

ทำได้ แต่ต้องทำการเปลี่ยน <u>Layout</u>

#### Hello Swing: Layout

#### Layout คือรูปแบบในการวาง Component แบบต่างๆ



#### Hello Swing: Layout

BorderLay	out 📃	1 🗶 🎯 G	ridLayout			- 0
Button 1			Button 1		2	
Button 3	2 Button	5	Button 3 Long-Named			tton 4
Long-N	amed Button 4		Button 5			
😹 FlowL	ayout					×
Butto	11 2	Button 3	Long-Named	Button	4 Button 5	
	BoxLayout		👹 GridBagLa	iyout		
	Button 1		Button 1	2	Button 3	
2 Button 3			Long-N	lamed B	utton 4	
<u> </u>	Long-Named But Button 5	ton 4		E	Button 5	

😹 CardLayout		
First card	Next card	card one
Previous card	Last card	

```
Hello Swing: Null Layout
```

}

```
public class Hello Swing {
public static void main(String[] args) {
JFrame F=new JFrame();
        F.setBounds(1, 1, 400, 200);
        JButton b1=new JButton("I am Button1");
        JButton b2=new JButton("I am Button2");
        JButton b3=new JButton("I am Button3");
        JPanel P=new JPanel();
        P.setLayout(null);
                                                                          \times
                                            <u></u>
        P.add(b1);
        P.add(b2);
        P.add(b3);
                                                Button หายไปไหนหมด ?
        F.setContentPane(P);
        F.show();
}
```

```
Hello Swing: Null Layout
public class Hello_Swing {
```

```
public static void main(String[] args) {
  JFrame F=new JFrame();
    F.setBounds(1, 1, 400, 200);
    JButton b1=new JButton("I am Button1");
    JButton b2=new JButton("I am Button2");
    JButton b3=new JButton("I am Button3");
    JPanel P=new JPanel();
    P.setLayout(null);
```

```
P.add(b1);
P.add(b2);
P.add(b3);
b1.setBounds(1, 1, 120, 25);
b2.setBounds(121, 1, 120, 25);
b3.setBounds(250, 120, 120, 25);
F.setContentPane(P);
F.show();
```

}

}

<u>\$</u>		_		$\times$
I am Button1	I am Button2			
		Lar	n Buttor	13
		lan	n Buttor	13

#### Hello Swing: Null Layout

I am Button1 I am Button2	I am Buttor	3						
			<u>\$</u>				_	×
			I am Button1	I am Button2				
					I am Button3	ขยายขนาดหน้าจอ		
I am Button I am Button	□ ×							
ยอขนาดหน้าจอ	la							

#### Hello Swing: Layout

#### Layout คือรูปแบบในการวาง Component แบบต่างๆ



#### Hello Swing: Grid Layout

public class Hello\_Swing {

```
public static void main(String[] args) {
   JFrame F=new JFrame();
        F.setBounds(1, 1, 400, 200);
        JButton b1=new JButton("I am Button1");
        JButton b2=new JButton("I am Button2");
        JButton b3=new JButton("I am Button3");
        JPanel P=new JPanel();
        GridLayout gLayout=new GridLayout(2,3);
        P.setLayout(gLayout);
    }
}
```

- P.add(b1);
  P.add(b2);
- P.add(b3);

}

}

F.setContentPane(P);
F.show();

1	2	3
4	5	6



#### Hello Swing: Grid Layout

```
public class Hello_Swing {
```

```
public static void main(String[] args) {
   JFrame F=new JFrame();
      F.setBounds(1, 1, 400, 200);
      JButton b1=new JButton("I am Button1");
      JButton b2=new JButton("I am Button2");
      JButton b3=new JButton("I am Button3");
      JPanel P=new JPanel();
      GridLayout gLayout=new GridLayout(2,3);
      P.setLayout(gLayout);
```

```
P.add(b1);
P.add(b2);
P.add(new JLabel(""));
P.add(new JLabel(""));
P.add(new JLabel(""));
P.add(b3);
```

```
F.setContentPane(P);
F.setContentPane(P);
```

}



1	2	3
4	5	6



#### Hello Swing: Grid Layout

🔮 – 🗆 X	
I am Button1 I am Button2	
<u>لا</u>	
I am Button3	
I am Button1 I am Butt	n2
I am Button1 I am Button2	
ลดขนาดจอ เล่า มี สิท Button 3 ขยายขนาดจอ	
	I am I



```
โปรแกรมแปลงค่าอุณหภูมิ
public class HelloTemperature {
public static void main(String[] args) {
 JFrame F=new JFrame("Hello Temperature");
 JPanel P=new JPanel();
 F.setBounds(1, 1, 500, 100);
 GridLayout gLayout=new GridLayout(1,3);
 P.setLayout(gLayout);
 F.add(P);
 JTextField t1=new JTextField();
 JButton b1=new JButton("Convert to Fahrenheit");
     JLabel 11=new JLabel("Hello");
     P.add(t1);
     P.add(b1);
     P.add(11);
                                    실 Hello Temperature
                                                                        \times
     F.show();
                                                    Convert to Fahrenheit
                                                                Hello
```

```
โปรแกรมแปลงค่าอุณหภูมิ
```

}

```
public class HelloTemperature {
public static void main(String[] args) {
 JFrame F=new JFrame("Hello Temperature");
 JPanel P=new JPanel();
 F.setBounds(1, 1, 500, 100);
GridLayout gLayout=new GridLayout(1,3);
 P.setLayout(gLayout);
 F.add(P);
 JTextField t1=new JTextField();
 JButton b1=new JButton("Convert to Fahrenheit");
     JLabel l1=new JLabel("Hello");
     P.add(t1);
     P.add(b1);
     P.add(11);
     b1.addMouseListener(new MouseAdapter(){
@Override
public void mouseClicked(MouseEvent arg0) {
Double c=Double.valueOf( t1.getText() );
Double f=(c^{*}(9/5.0)+32);
11.setText(String.format("%.2f'F", f) );
                                               🛓 Hello Temperature
                                                                                               ×
                                                                    Convert to Fahrenheit
                                              37
                                                                                     98.60'F
     });
     F. show();
}
```

#### Swing: Windows Application

# สรุปแล้วโครงสร้างพื้นฐานของ GUI จะประกอบด้วย

3 ส่วนนี้ต้องใช้คู่กันเสมอ

ตั้งแต่แรก ?

แล้วทำไมไม่รวม หรือ ใส่ให้เองเลย

- 1) JFrame
- 2) JPanel
- 3) Layout
- 4) Component
- 5) Event Handler (Event listener)

จริงๆ swing framework ทำการติดตั้ง ส่วนประกอบทั้ง 3 ไว้ใน JFrame แล้ว และยังมี ส่วนประกอบอื่นที่เรายังไม่ได้ใช้งาน



#### Swing: Windows Application



## การบ้าน 5

ให้ออกแบบหน้าจอของโปรแกรมแปลงเงินบาทเป็นสกุลอื่นประกอบไปด้วย USD, EUR, GBP, JPY และ CNY <u>ที่เหมาะสมกับการใช้งานจริง</u> โดยกำหนดให้เลือกใช้ component ดังต่อไปนี้



เขียนคำตอบใส่<u>กระดาษ A4 จำนวน 1 แผ่น</u>ส่งทางกล่องรับการบ้าน หน้าแลป 2 เขียนชื่อ รหัส กลุ่มเรียน ที่<u>มุมบนขวา</u> ภายในวันศุกร์ที่ 8 กันยายน 2560