

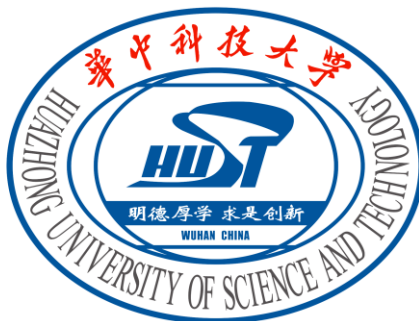
基于Java的面向对象程序设计

陈维亚

weiya_chen@hust.edu.cn

华中科技大学软件学院

第25-26讲：Java图形编程



1. GUI简介
2. Swing 组件
3. 布局器
4. 交互
5. 2D graphics

□ Java Foundation Class (JFC)

是否应该掌握Java 图形库？

□ Java Foundation Class (JFC)

➤ AWT (Abstract Windowing Toolkit)

- 为Java的成功发挥了重要作用
- 重量级组件，依赖于系统的底层图形实现
- Swing直接依赖于AWT（事件、布局管理等）

➤ Swing

- 提供了更多的组件和完善的功能
- 轻量级组件，100% Java代码实现

➤ 通常，我们使用

- Swing组件组合界面
- AWT进行事件响应和处理
- AWT的布局管理器完成组件的布局

➤ 注意区分组件名称：

- AWT: Frame, Button, Panel, ...
- Swing: JFrame, JButton, JPanel, ...

1. GUI 简介



□ AWT 库

Component

显示单元，事件源

paint
addXXXListener

```
public abstract class Component  
    extends Object  
    implements ImageObserver,  
    MenuContainer, Serializable
```

直接子类：

Button, Canvas, Checkbox, Choice, Container, Label, List, Scrollbar, TextComponent

1. GUI 简介



□ AWT 库

Component

显示单元，事件源

paint
addXXXListener

Container

包含其它组件，
管理它们的位置

add
setLayout

```
public class Container  
    extends Component
```

直接子类：

JComponent, Panel, ScrollPane, Window 等

1. GUI 简介



□ AWT 库

Component

显示单元，事件源

paint
addXXXListener

Container

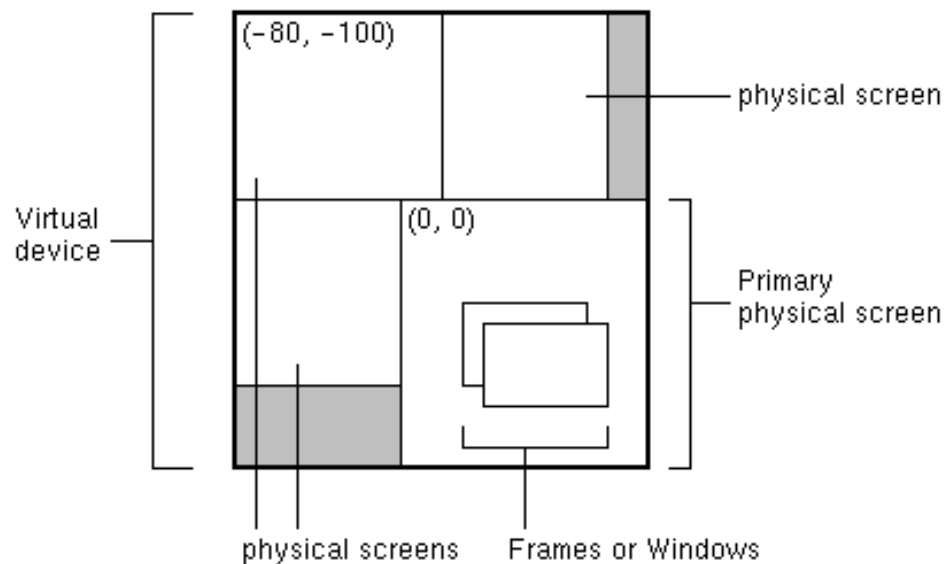
包含其它组件，
管理它们的位置

add
setLayout

Window

最外层组件，既无边框，也无菜单

setVisible
setOwner



1. GUI 简介



□ AWT 库

Component

显示单元，事件源

paint
addXXXListener

Container

包含其它组件，
管理它们的位置

add
setLayout

Window

最外层组件，既无边框，也无菜单

setVisible
setOwner

Frame

包含窗口主体内容(标题、边框)

直接子类：

JFrame

1. GUI 简介



□ AWT 库

Component

显示单元，事件源

paint
addXXXListener

Container

包含其它组件，
管理它们的位置

add
setLayout

Window

最外层组件，既无边框，也无菜单

setVisible
setOwner

Frame

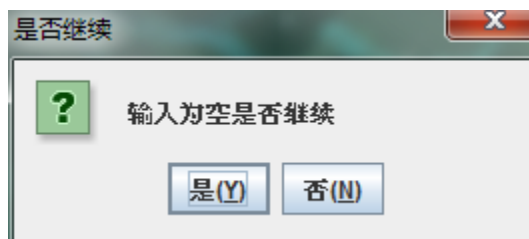
包含窗口主体内容(标题、边框)

Dialog

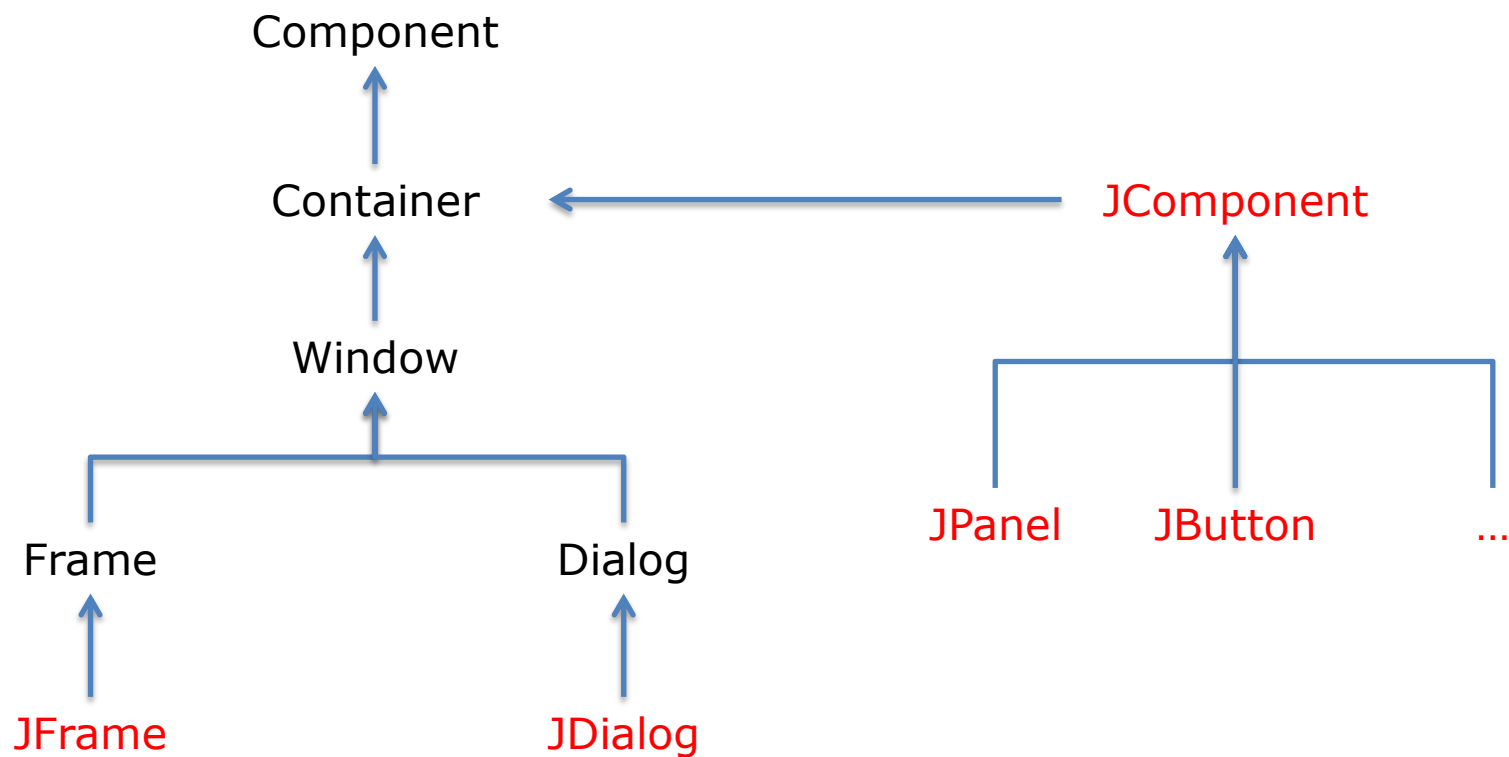
临时性的对话框

直接子类：

FileDialog, JDialog



□ AWT + Swing



□ GUI HelloWorld

```
import javax.swing.JFrame;

public class Simple1 {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Hello World");
        frame.setSize(300, 200);
        frame.setVisible(true);
    }
}
```



点击关闭图标后，
程序并未结束

□ GUI HelloWorld

```
import javax.swing.JFrame;

public class Simple2 {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Hello World");
        frame.setSize(300, 200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

□ GUI HelloWorld

```
import javax.swing.JFrame;  
import javax.swing.JPanel;  
import java.awt.Color;  
  
public class Simple3 {  
    public static void main(String[] args) {  
        JFrame frame = new JFrame("Hello World");  
        frame.setSize(300, 200);  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  
        JPanel panel = new JPanel();  
        panel.setBackground(Color.red);  
        frame.setContentPane(panel);  
        frame.setVisible(true);  
    }  
}
```



□ GUI HelloWorld 另一种写法

```
import javax.swing.JFrame;
import javax.swing.JPanel;
import java.awt.Color;

public class Simple4 extends JPanel {
    Simple4() {
        JLabel label = new JLabel("Bonjour!");
        label.setForeground(Color.white);
        label.setFont(new Font("Serif", Font.PLAIN, 48));
        this.add(label);
        this.setBackground(Color.red);
    }

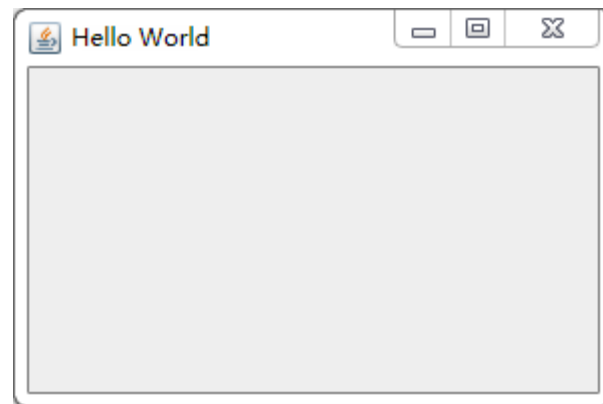
    public static void main(String[] args) {
        JFrame frame = new JFrame("Hello World");
        frame.setSize(300, 200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setContentPane(new Simple4());
        frame.setVisible(true);
    }
}
```

□ 顶层组件

包括 JFrame、JDialog、JApplet、JWindow

JFrame是大多数应用程序的基本窗口，有边框、标题和按钮，允许程序员把其他组件添加到它里面，把它们组织起来，并把它们呈现给用户。

```
JFrame frame = new JFrame("Hello World");  
  
// Set properties  
...  
// Add components  
...  
frame.pack();  
frame.setVisible(true);
```



□ 中间组件 - 二级容器

必须放在顶层容器内，且能够容纳其他控件

包括JPanel、JScrollPane、JToolBar、JSplitPane、JTabbedPane

- JPanel：常规面板；
- JScrollPane：滚动面板，即带有长宽滚动条，主要用来容纳大型控件；
- JToolBar：工具栏面板，包含图标按钮；
- JSplitPane：分割式面板；
- JTabbedPane：选项卡面板；

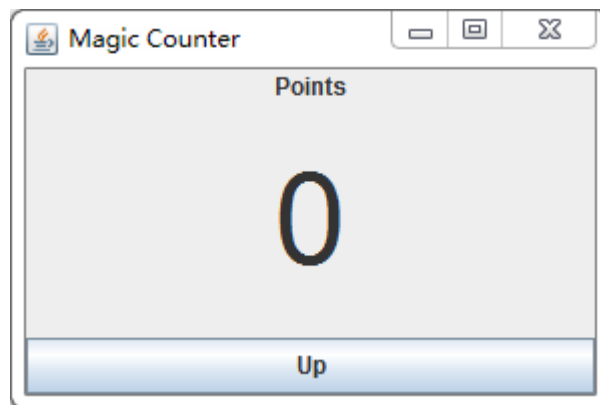
□ 基本组件

继承自JComponent的组件都是swing的基本组件，它们有一些通用的方法，如设置边框、背景颜色等。

- **按钮**：JButton（常规按钮）、JCheckBox（复选框）、JRadioButton（单选按钮）。
- **文本组件**：JTextField（文本字段）、JPasswordField（密码框）、JTextArea（纯文本）。
- **不可编辑显示组件**：JLabel（显示不可编辑文本）、JToolTip（显示不可编辑文本）、JProgressbar（进度条）。
- **菜单**：JMenu（普通菜单）、JPopupMenu（弹出式菜单）。
- **其他组件**：JFileChooser（文件选择器）、JColorChooser（颜色选择器）、JTable（表格）、JTree（树）、JComboBox（下拉框）等等。

□ 实现一个计数器

设计一个如下图所示的计数器面板，需要用到 JPanel , JLabel 和 JButton 组件。



3. 布局器

□ 机制



Boîte de saisie

Nom ☐ Tennis

Prenom ☐ Squash

Adresse ☐ Natation

☐ Athlétisme

☐ Randonnée

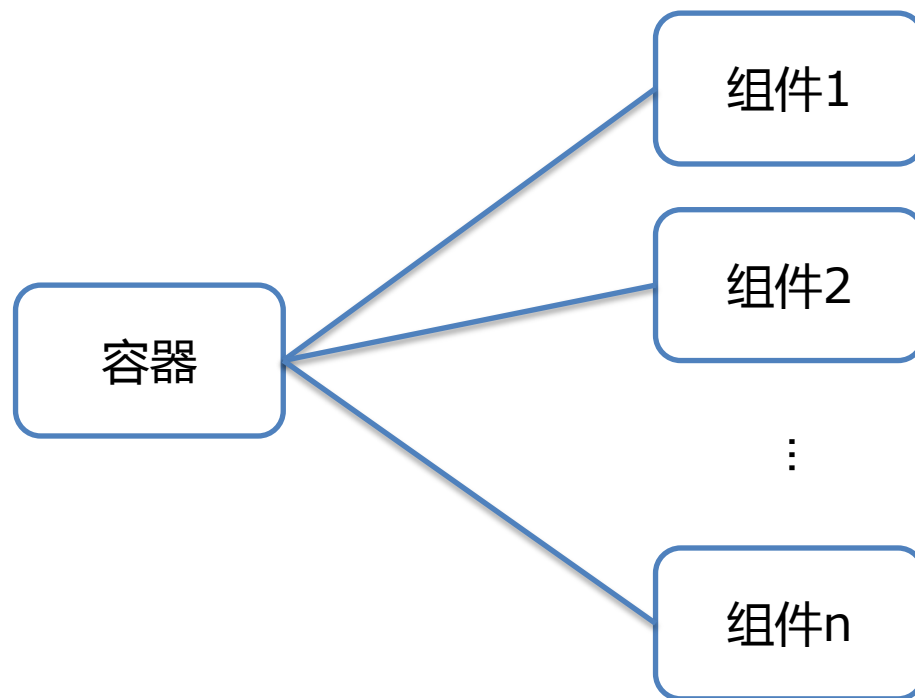
☐ Foot

☐ Basket

☐ Volley

Sexe ☒ Homme ☐ Femme ☐ Petanque

OK Annuler



3. 布局器

□ 机制



Boîte de saisie

Nom ☐ Tennis

Prenom ☐ Squash

Adresse ☐ Natation

☐ Athlétisme

☐ Randonnée

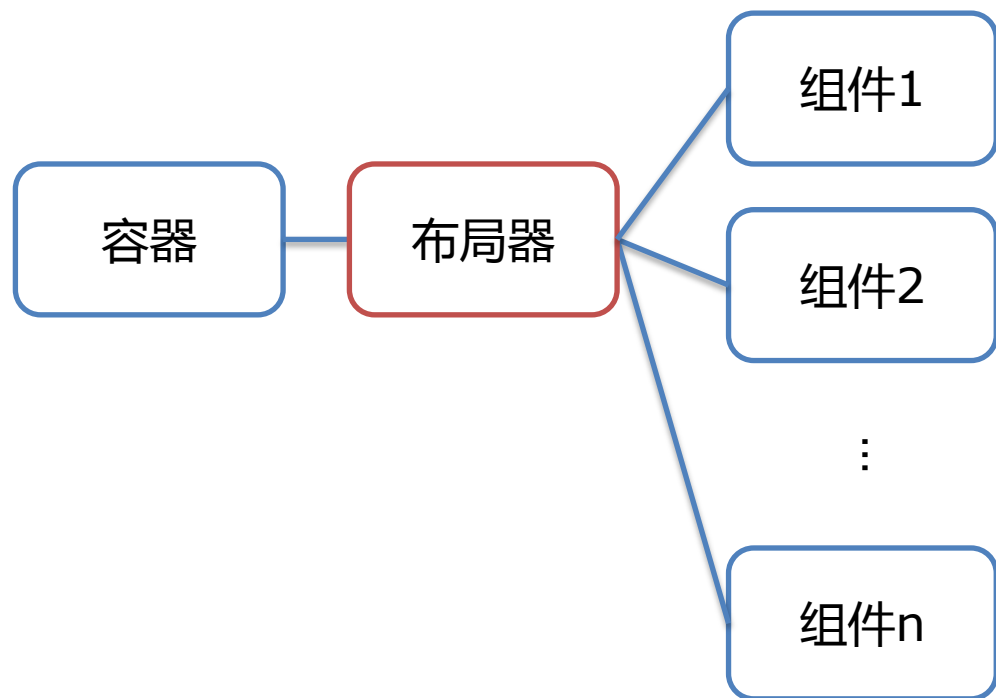
☐ Foot

☐ Basket

☐ Volley

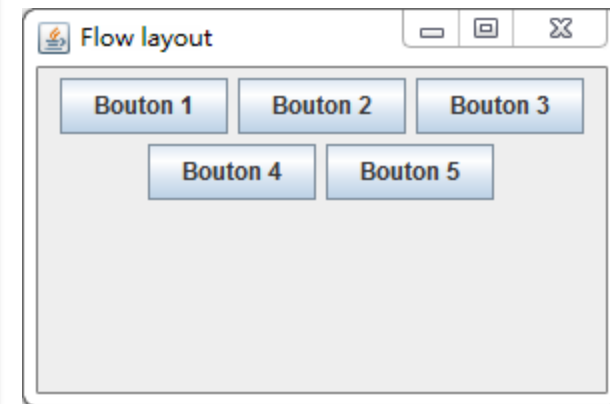
Sexe ☒ Homme ☐ Femme ☐ Petanque

OK Annuler



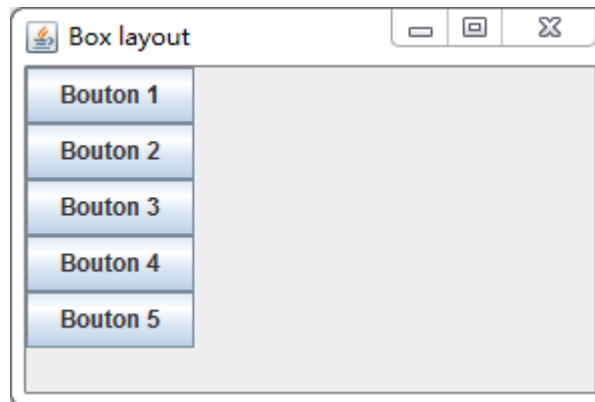
□ Flow layout

```
public class Test extends JPanel {  
    Test() {  
        setLayout(new FlowLayout());  
        for (int i = 1; i <= 5; i++)  
            add(new JButton("Bouton " + i));  
    }  
  
    public static void main(String[] args) {  
        JFrame cadre = new JFrame("Flow layout");  
        cadre.setSize(300, 200);  
  
        cadre.setDefaultCloseOperation(JFrame.EXIT_ON_C  
LOSE);  
        cadre.setContentPane(new Test());  
        cadre.setVisible(true);  
    }  
}
```



❑ Box layout

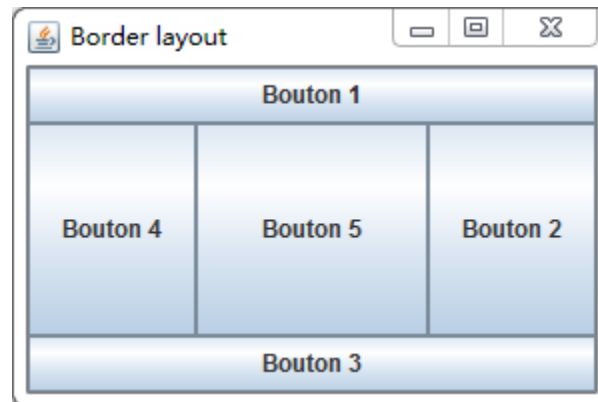
```
setLayout(new BorderLayout(this, BorderLayout.Y_AXIS));  
for (int i = 1; i <= 5; i++)  
    add(new JButton("Bouton " + i));
```



❑ Border layout

```
setLayout(new BorderLayout());
```

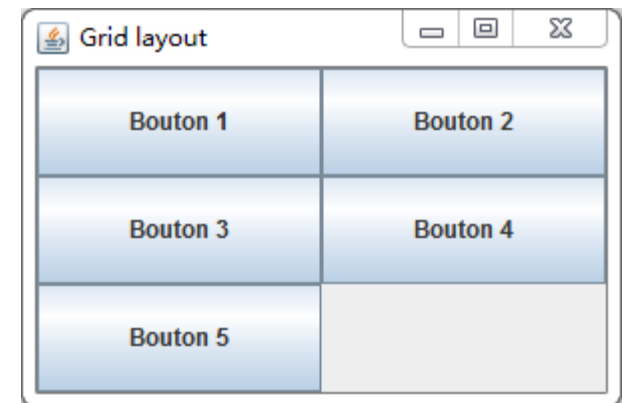
```
add(new JButton("Bouton 1"), BorderLayout.NORTH);  
add(new JButton("Bouton 2"), BorderLayout.EAST);  
add(new JButton("Bouton 3"), BorderLayout.SOUTH);  
add(new JButton("Bouton 4"), BorderLayout.WEST);  
add(new JButton("Bouton 5"), BorderLayout.CENTER);
```



3. 布局器

□ Grid layout

```
setLayout(new GridLayout(3, 2));  
for (int i = 1; i <= 5; i++)  
    add(new JButton("Bouton " + i));
```



3. 布局器

□ 组合嵌套



Boîte de saisie

Nom

Prenom

Adresse

☐ Tennis

☐ Squash

☐ Natation

☐ Athlétisme

☐ Randonnée

☐ Foot

☐ Basket

☐ Volley

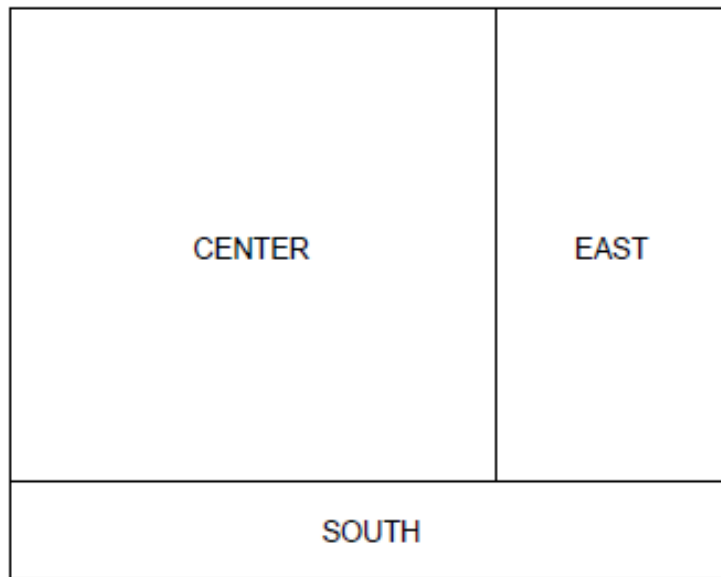
☐ Petanque

Sexe ☒ Homme ☐ Femme

OK Annuler

3. 布局器

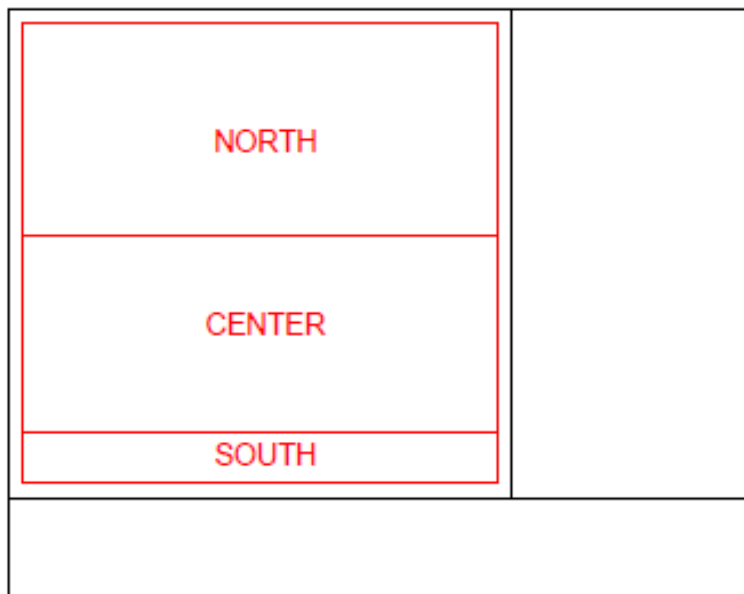
□ 组合嵌套



The screenshot shows a Java Swing dialog box titled "Boîte de saisie". The dialog box contains several input fields and a list of checkboxes. The input fields are labeled "Nom", "Prenom", and "Adresse". The "Adresse" field is a text area. Below the input fields are radio buttons for "Sexe" with options "Homme" (selected) and "Femme". To the right of the input fields is a list of checkboxes for sports: "Tennis", "Squash", "Natation", "Athlétisme", "Randonnée", "Foot", "Basket", "Volley", and "Petanque". At the bottom of the dialog box are two buttons: "OK" and "Annuler".

3. 布局器

□ 组合嵌套



The screenshot shows a Java Swing dialog box titled "Boîte de saisie". It contains several input fields and a list of checkboxes. A red rectangle highlights the "Nom", "Prenom", and "Adresse" fields, along with the "Sexe" label and radio buttons. The "Sexe" section has "Homme" selected. The list of checkboxes on the right includes "Tennis", "Squash", "Natation", "Athlétisme", "Randonnée", "Foot", "Basket", "Volley", and "Petanque". The dialog has "OK" and "Annuler" buttons at the bottom.

□ 实现一个计数器

运用不同的布局器组件扩展之前实现的计数器。

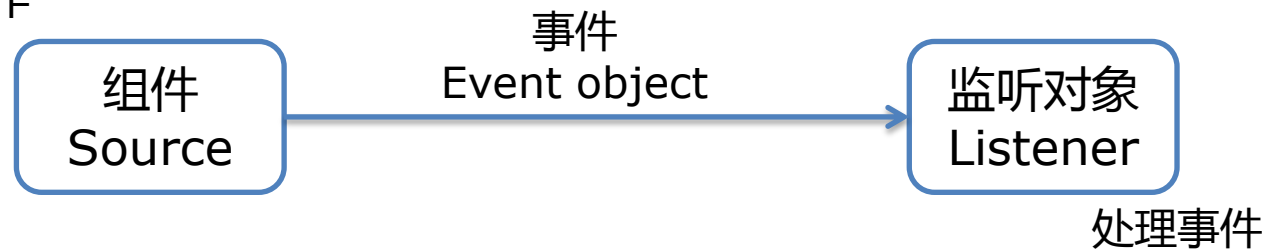


□ 实现一个电子时钟

运用线程实现一个自动刷新的数字时钟。

□ 事件处理机制

用户操作



□ 事件处理接口

➤ ActionListener 接口

```
public interface ActionListener  
extends EventListener
```

Method	Purpose
<code>actionPerformed(ActionEvent)</code>	Called just after the user performs an action.

➤ ActionEvent 类

```
public class ActionEvent  
extends AWTEvent
```

Method	Purpose
<code>String getActionCommand()</code>	Returns the string associated with this action. Most objects that can fire action events support a method called <code>setActionCommand</code> that lets you set this string.
<code>int getModifiers()</code>	Returns an integer representing the modifier keys the user was pressing when the action event occurred.
<code>Object getSource()</code>	Returns the object that fired the event.

➤ ActionListener 示例

```
public class Beeper implements ActionListener {  
    ...  
    // where initialization occurs:  
    button.addActionListener(this);  
  
    public void actionPerformed(ActionEvent e) {  
        //Make a beep sound...  
    }  
}
```


➤ MouseListener 接口

```
public interface MouseListener  
extends EventListener
```

```
void mousePressed(MouseEvent evt);  
void mouseReleased(MouseEvent evt);  
void mouseClicked(MouseEvent evt);  
void mouseEntered(MouseEvent evt);  
void mouseExited(MouseEvent evt);
```

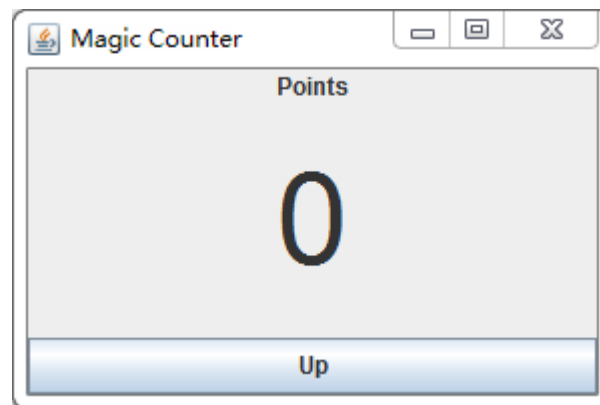
```
source.addMouseListener(mListener);
```

完整 Listener API :

<https://docs.oracle.com/javase/tutorial/uiswing/events/api.html>

□ 实现一个计数器

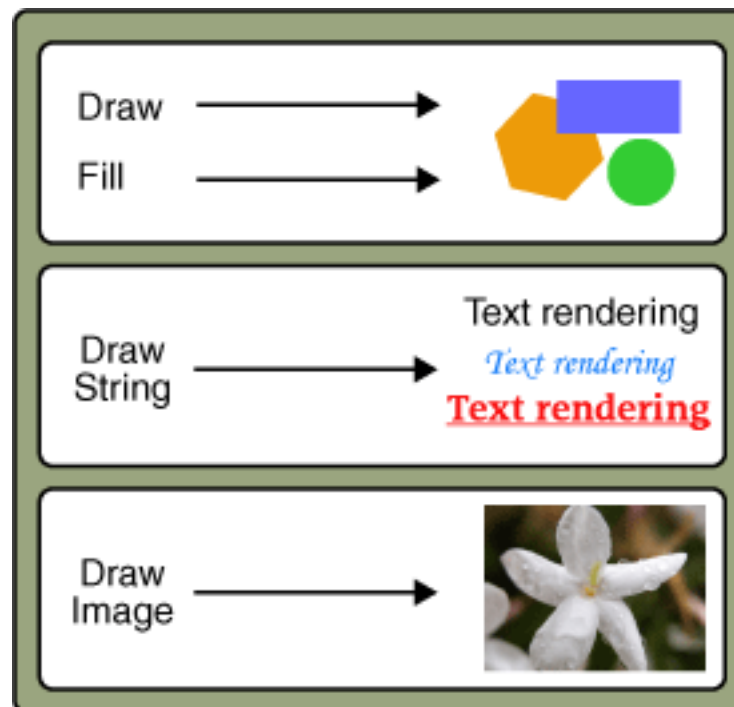
完成计数器，当用户点击按钮的时候，计数器显示的数字+1。



5. 2D graphics

自定义 JPanel 的子类，重写 paint() 方法实现绘图

```
public void paint(Graphics g)
```



实验如下的画线程序（自定义Point类）：

```
public class Painter extends JPanel implements MouseListener {
    ArrayList<Point> points = new ArrayList<Point>();

    public void mousePressed(MouseEvent e) {
        points.add(new Point(e.getX(), e.getY()));
        repaint();
    }

    public void paint(Graphics g) {
        super.paint(g);
        Iterator<Point> iter = points.iterator();
        if (iter.hasNext()) {
            Point pr = iter.next();
            while (iter.hasNext()) {
                Point pc = iter.next();
                g.drawLine(pr.x, pr.y, pc.x, pc.y);
                pr = pc;
            }
        }
    }
}
```

接上

```
public void mouseReleased(MouseEvent e) { }
public void mouseClicked(MouseEvent e) { }
public void mouseEntered(MouseEvent e) { }
public void mouseExited(MouseEvent e) { }

Painter() {
    setBackground(Color.white);
    addMouseListener(this);
}

public static void main(String[] args) {
    JFrame cadre = new JFrame("Line Painter");
    cadre.setSize(600, 400);
    cadre.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    cadre.setContentPane(new Painter());
    cadre.setVisible(true);
}
}
```

Java 图形库

GUI 布局

交互（事件处理机制）



GUI 的面向对象设计

如何组合GUI与业务逻辑类

Java 数据库