Chapter 8
Java and Graphical User Interface

Wang Yang
wyang AT njnet.edu.cn

Outline

- Multi-thread Review
- Introduction of Java GUI Foundation
- Elements of Java GUI
- Layout of Java GUI
- Event of Java GUI
- Reference

Multi-thread Review

Thread & Runnable

- Thread Class
- Runnable Interface
- run() method
- control thread
  - start / join / interrupt

Thread Safe

- synchronized method
- synchronized statement
- the lock belong to object

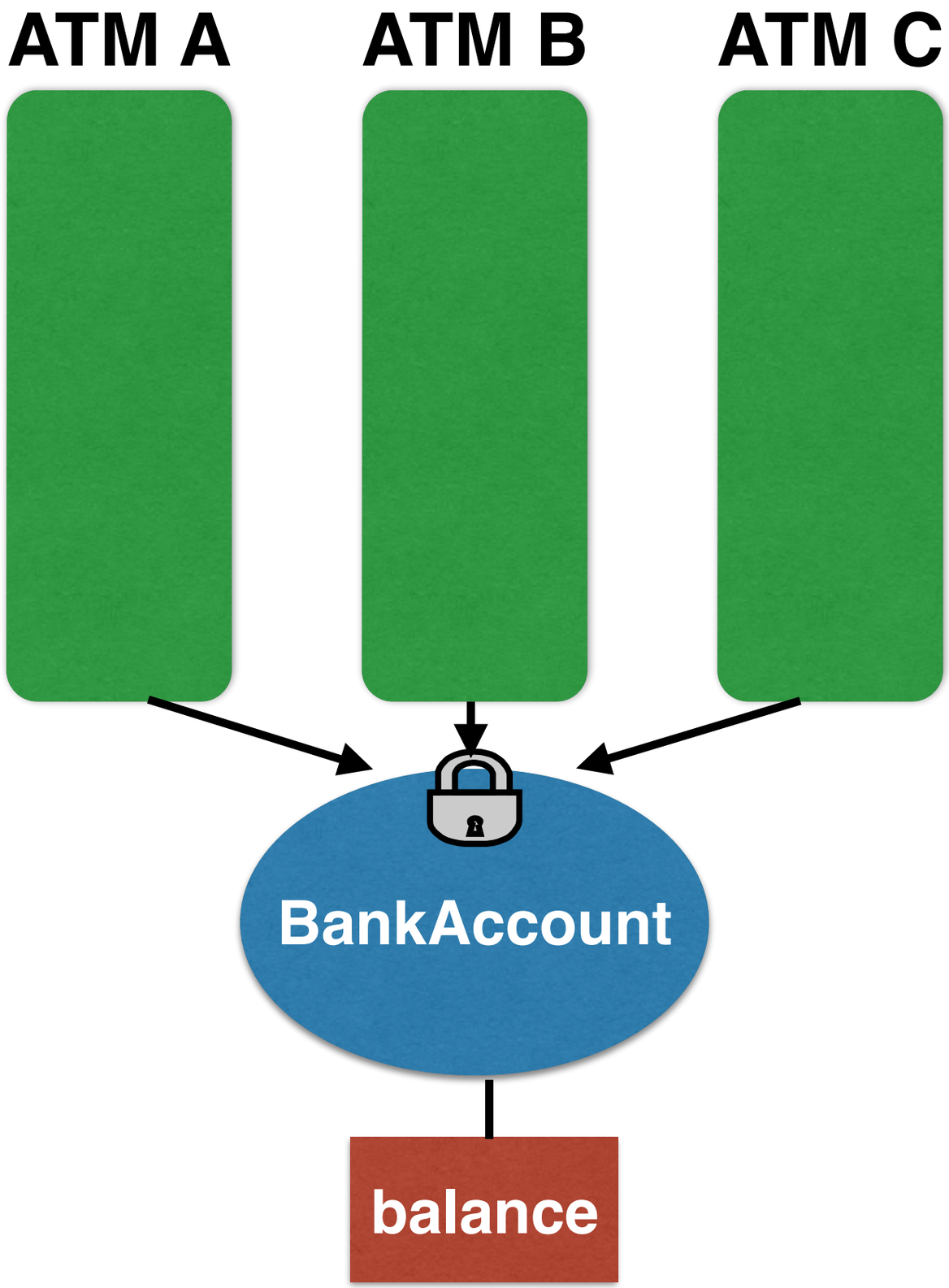
Synchronization Method

```
public class BankAccount {
    private long id;
    private long balance;
    public BankAccount(){
        balance = 0;
    }
    public synchronized long getBalance(){
        return balance;
    }
    public synchronized void deposit(long amount){
        balance += amount;
    }
```

Synchronization Method

```
public class ATM extends Thread{
    private BankAccount account;
    public ATM(BankAccount account){
        this.account = account;
    }
    public void run(){
        for(int i = 0; i < 10000; i++)
            account.deposit(10);
    }
}
```

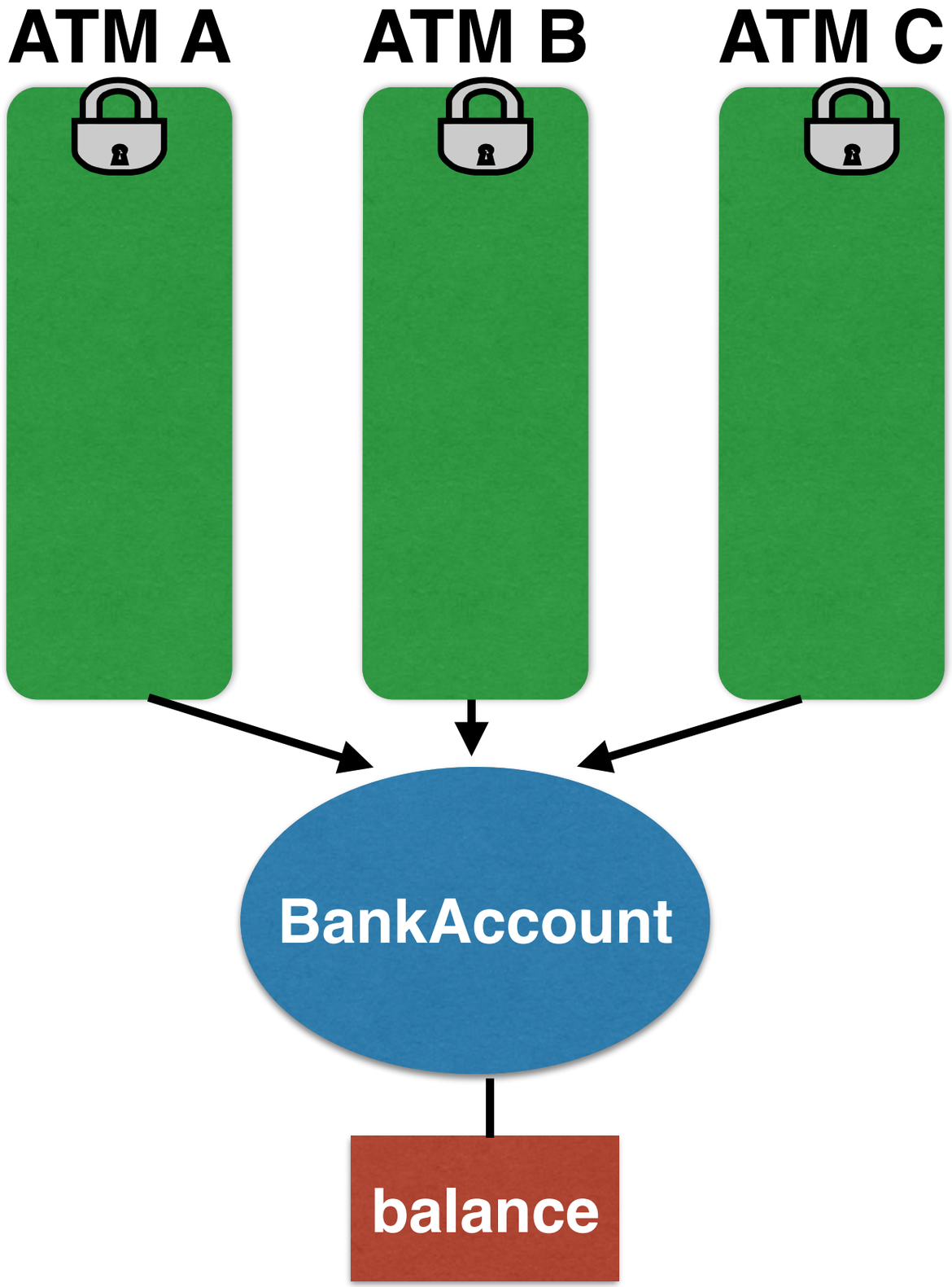
Synchronization Method



Right

```
blic class ATM extends Thread{
   private BankAccount account;
   public ATM(BankAccount account){
        this.account = account;
   }                          ???????
   public synchronized void run(){
        for(int i = 0; i < 10000; i++)
            account.deposit(10);
   }
```

Synchronization Method



Wrong

Ticket Problem

- Which should be locked?

- 10 Station

- 1 ticket resource

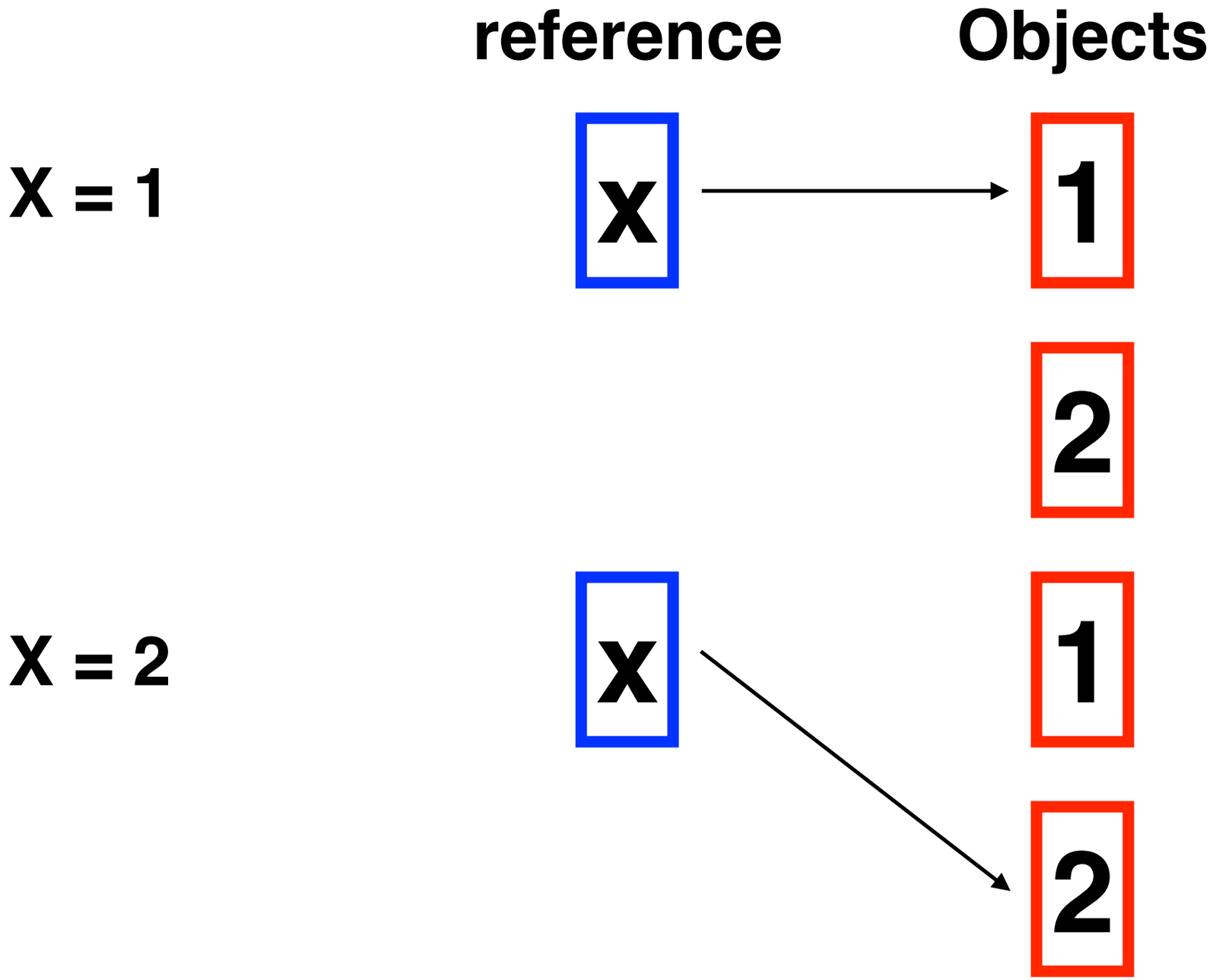
Ticket Problem

- how to lock the ticket resource
  - make it a object

```
public class Ticket{
    int totalNum;
    int id;
}
```

immutant object

- mutant object
  - value of object can be changed
- immutant object
  - value of object can’t be changed
  - Integer, String
  - when you change the value, you change the object


reference
Objects
X = 1
x
1
2
X = 2
x
1
2

Introduction of Java

GUI Foundation

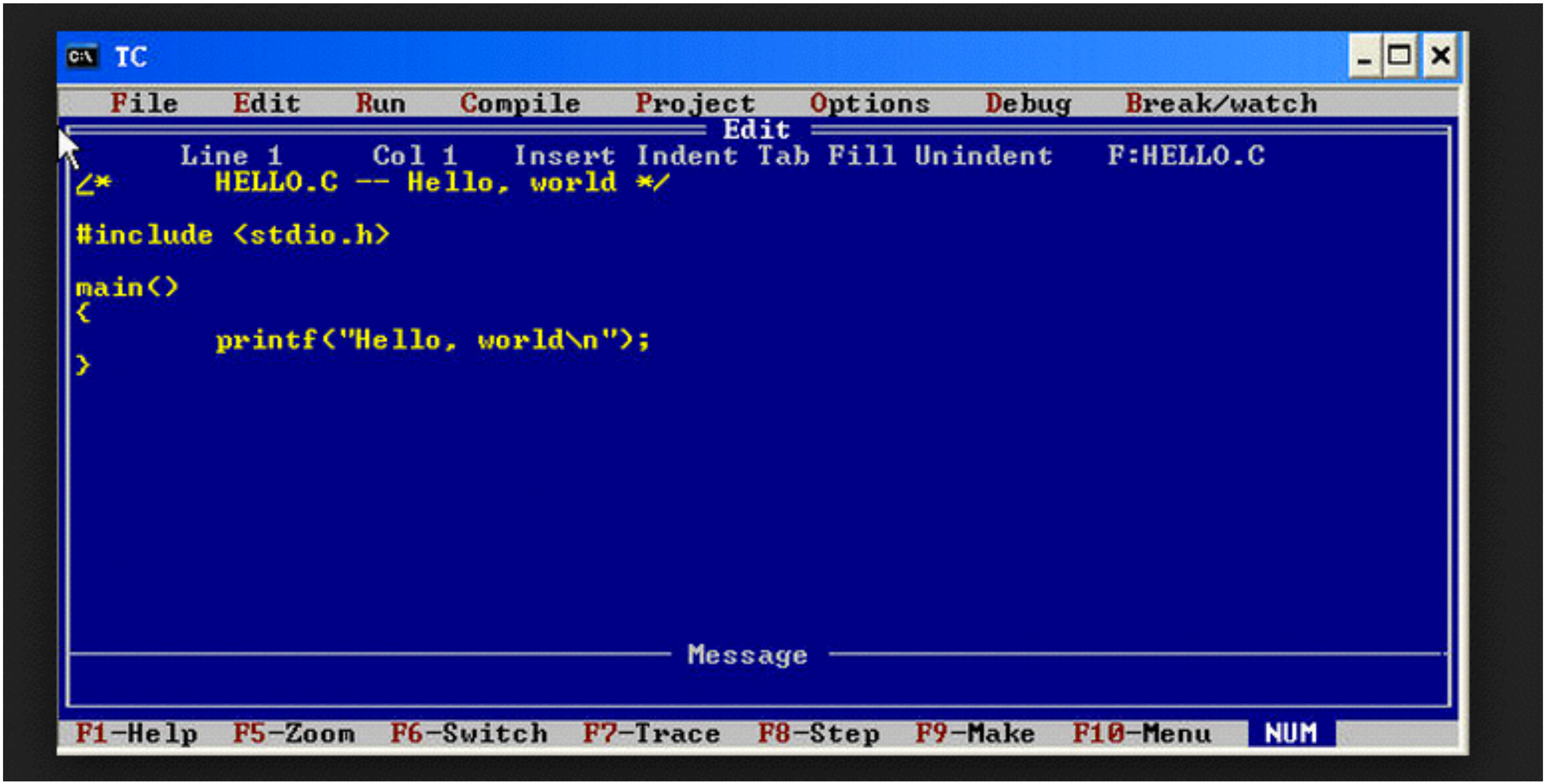
What is GUI

- Graphical User Interface
  - a type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators
  - opposed to text-based interfaces, typed command labels or text navigation.
  - beautiful, better learning curve
  - originate from Palo Alto (1981)

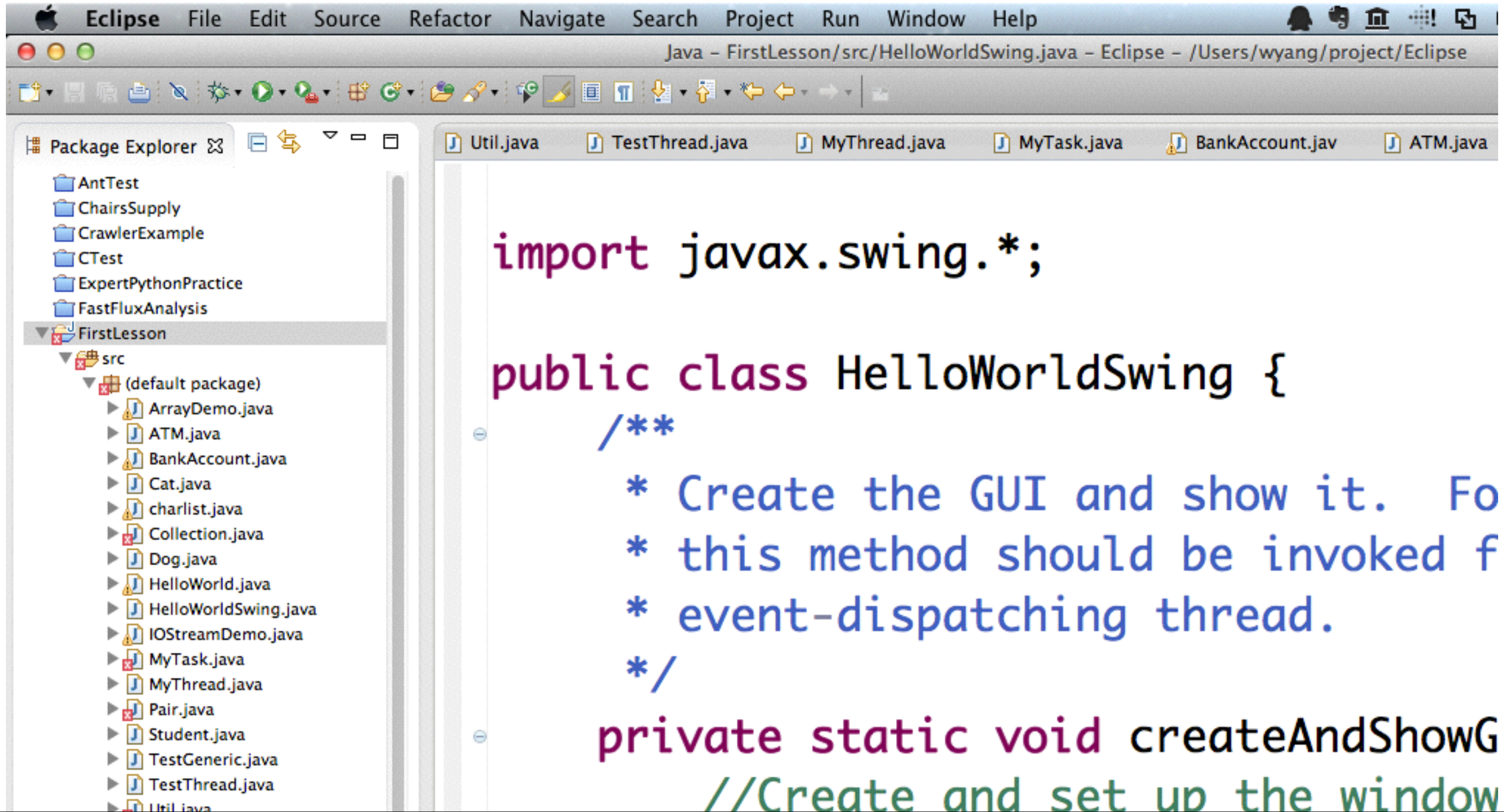
Command User Interface

```
wyangtekiMacBook-Pro:~ wyang$ python
Python 2.7.6rc1 (v2.7.6rc1:4913d0e9be30+, Oct 27 2013, 20:52:11)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print "hello world"
hello world
>>>
```

Command User interface



Graphical User Interface



What GUI Contain

- Components
  - different components, Text, Label, Icon, Button..
- Layout
  - how to combine the component
- Action/Event
  - user interaction

How to get so many Components

- native component
  - use the components provided by the OS
  - remember controls in MFC?
    - but if you want a component the OS doesn’t support…
- emulated component
  - draw everything by the language library
  - you can get nearly everything you want
  - the performance….

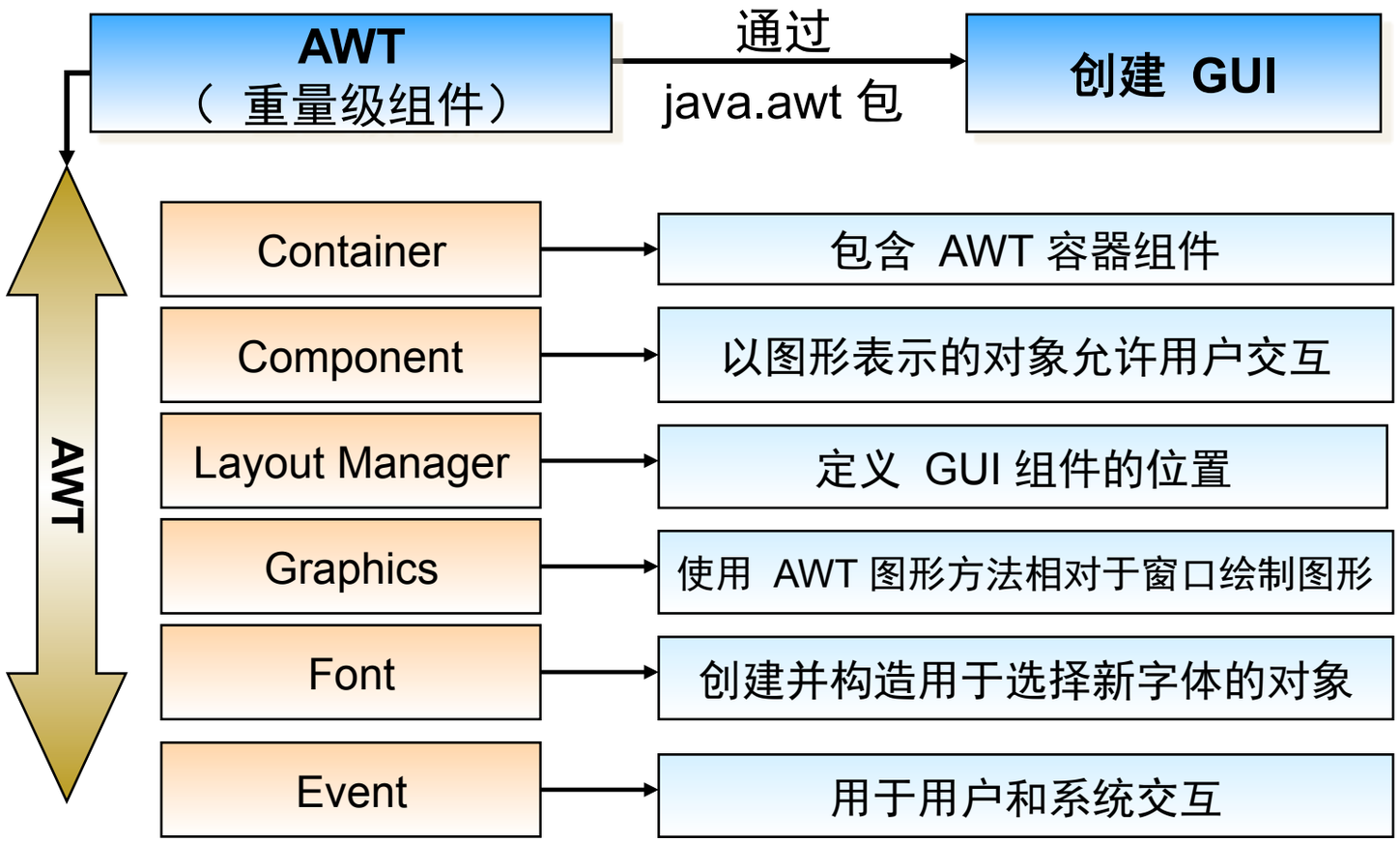
Two Camps

- Sun
  - AWT/Swing
    - I Support emulated
- IBM
  - SWT/JFace
  - I support native

AWT Introduction

- java.awt.*
  - java 1.0, it is long long ago…
  - Abstract Window Toolkit
  - Basic UI component of Java
  - developed in one month before the JDK1.0 published

AWT Introduction



AWT Introduction

- Early Technology of Java
  - Limited component
    - use Greatest Common Divisor to implement Components
  - Different appearance in different platform
    - native Component
    - Write Once, Different Everywhere
  - No pop-up menu, scrolling pane, keyboard navigation…

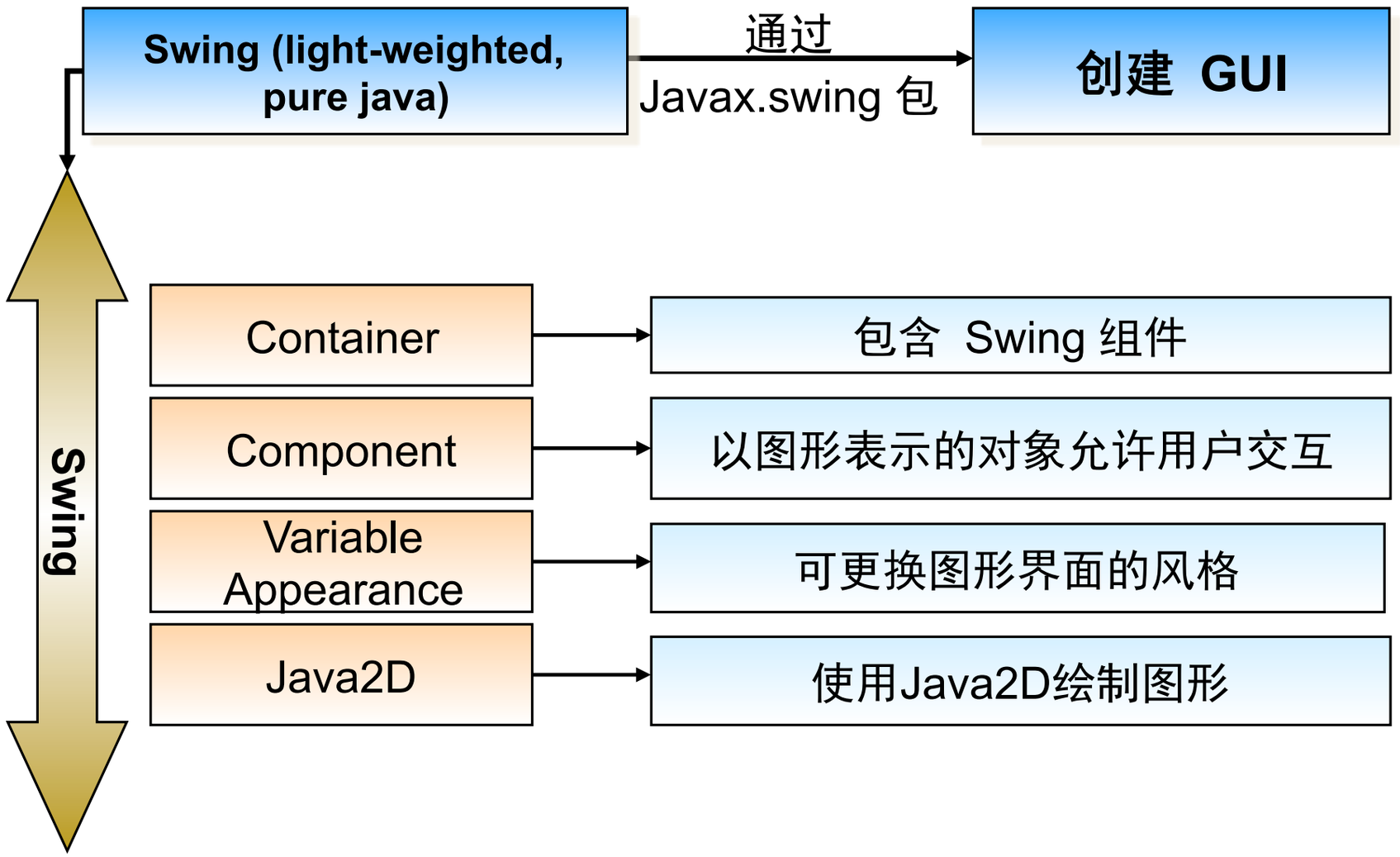
Swing Introduction

- Overcome AWTs Shortage
  - Pure Java
  - Swing package is based on AWT
  - Swing is slower than AWT
- javax.swing

Swing Introduction

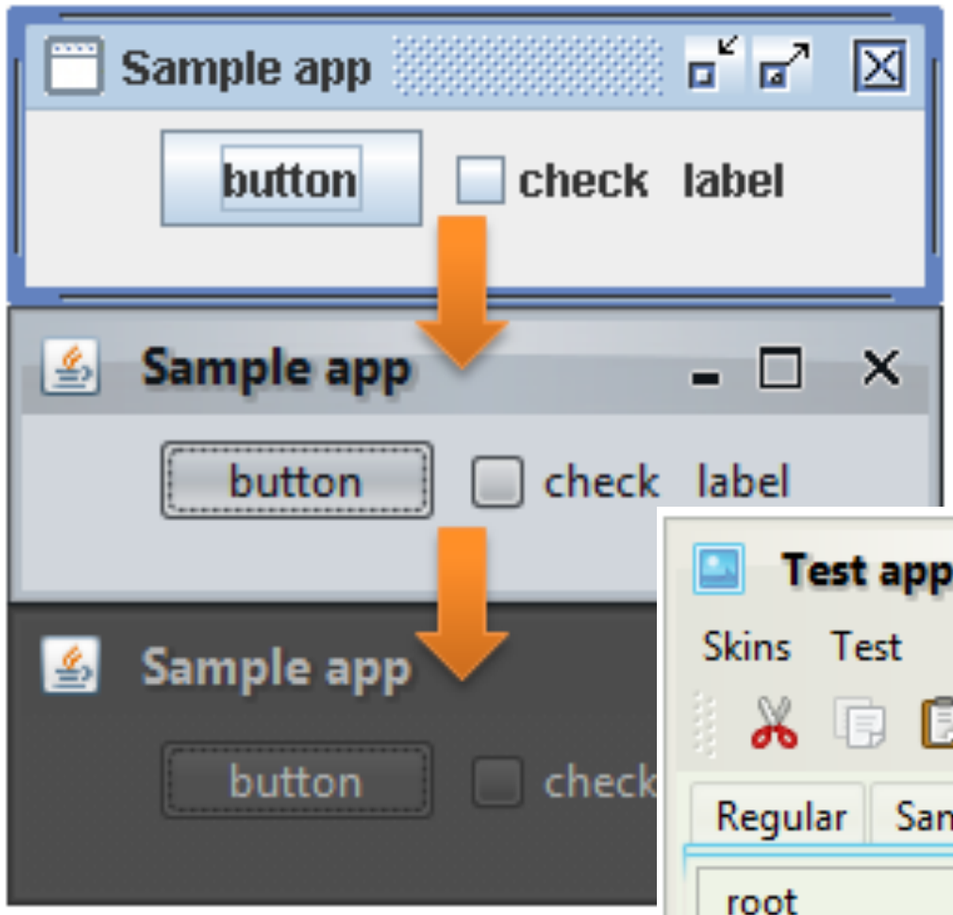
- Overcome AWTs Shortage
  - Pure Java
  - Swing package is based on AWT
  - Swing is slower than AWT
- javax.swing

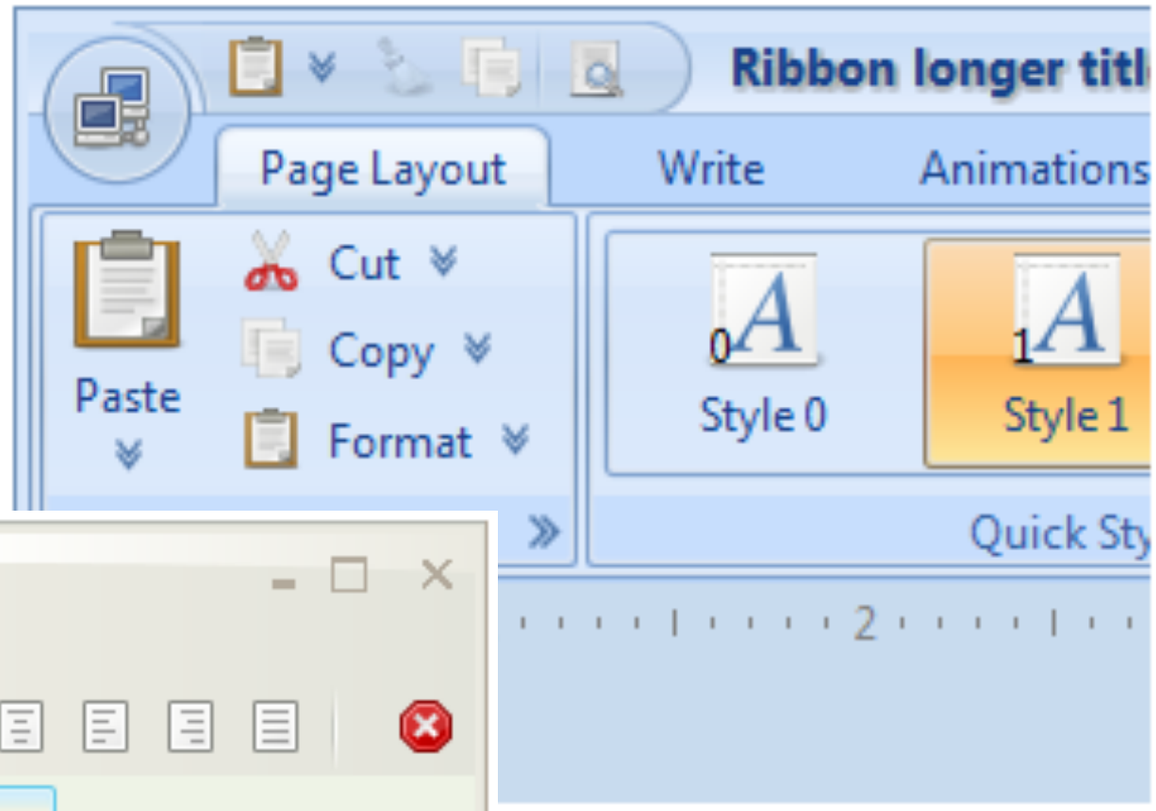
Swing Introduction

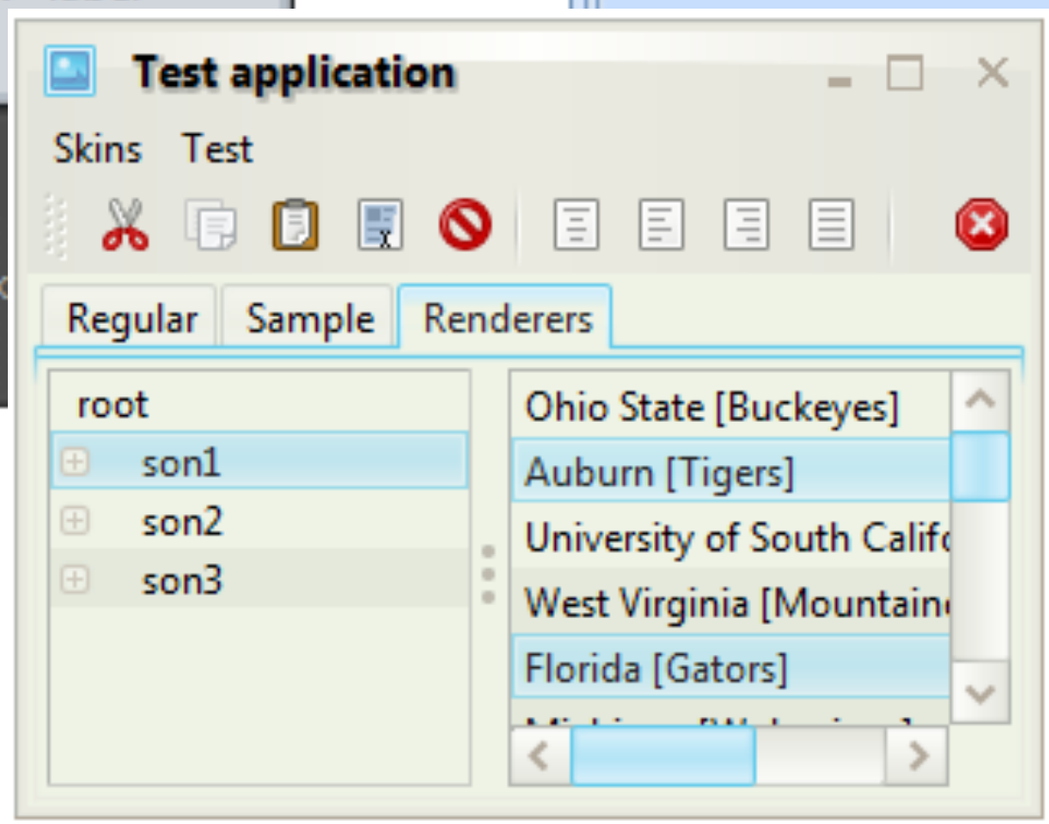


Swing - Appearance

- Motif / Windows / Mac / Custom







Swing Introduction

	AWT	Swing
Devoloper	Sun JDK	Sun JDK
Implemen-tation	Heavy-weighted ; GCD ; Invoke OS Component	Light-weighted ; Top-level container invoke OS component; most component is in pure java
Portablity	Appearance and Behavior depend on OS	Independent with OS
Speed	Fast	Slow before Jdk1.4, but faster now
Component	No abundant	Abundant
Visual Development	No	Jbuilder，Netbeans，Eclipse VE

SwingSet3

File View

GUI Components

View: by category

Toplevel Containers

JFrame

JDialog

JWindow

Containers

JInternalFrame

JTabbedPane

JScrollPane

JSplitPane

GridBagLayout

Data

JTable

JTree

JList

Controls

ToggleButtons

JButton

no code loaded

Java Application Window

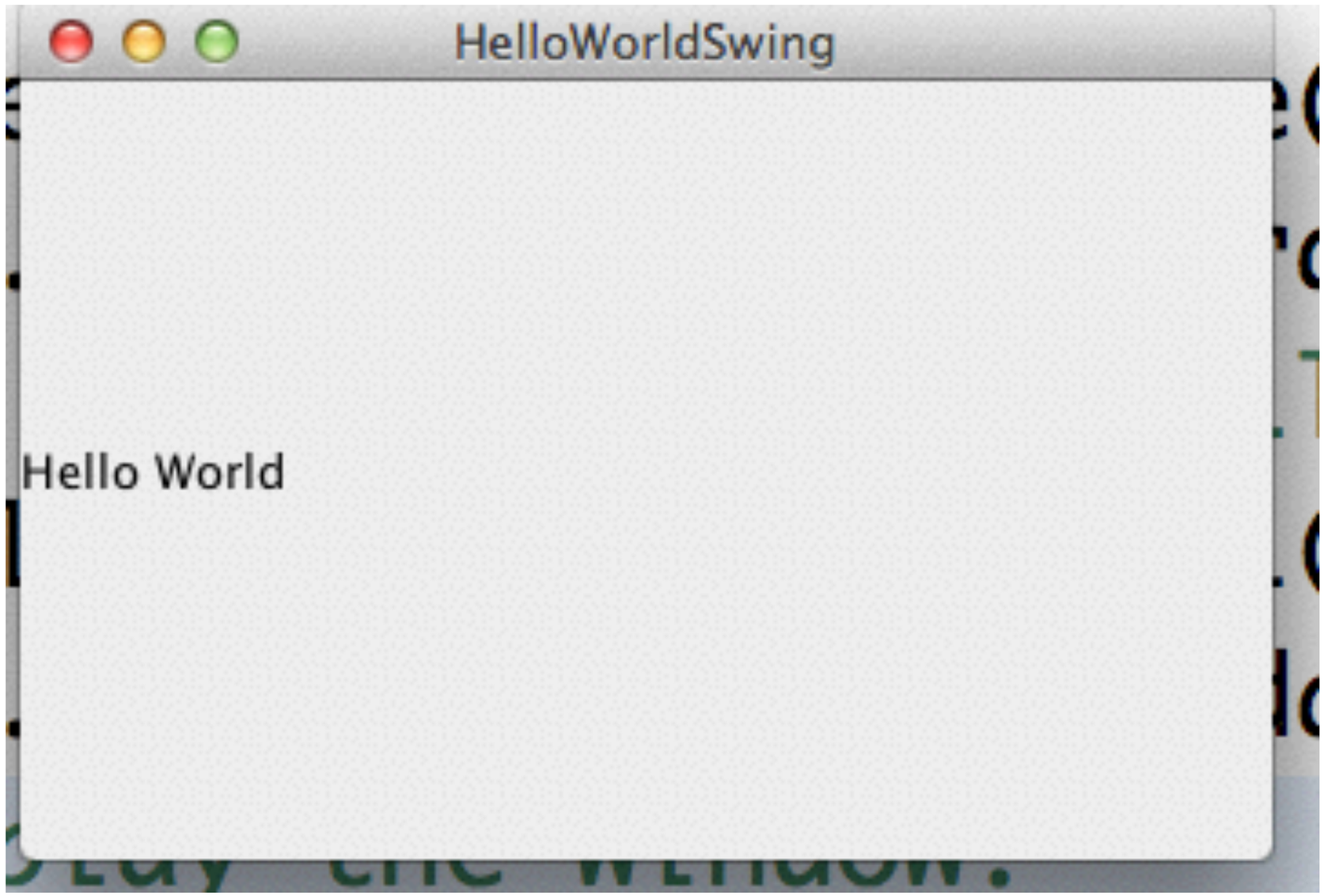
One More Word About SWT

- SWT / JFace
  - IBM published /
  - born with Eclipse
- native Component principle
- Least Common Multiple Principle
  - combine different os' components
  - if the os doesn't have the component, emulate it
  - else use the native component

A simple Swing Program

```
public class HelloWorldSwing {
    private static void createAndShowGUI() {
        //Create and set up the window.
        JFrame frame = new JFrame("HelloWorldSwing");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE)
        //Add the ubiquitous "Hello World" label.
        JLabel label = new JLabel("Hello World");
        frame.getContentPane().add(label);
        //Display the window.
        frame.setVisible(true);
    }
    public static void main(String[] args) {
        createAndShowGUI();
    }
```

A simple Swing Program



Components of Java GUI

What Element we have

- Container
  - contain different components
  - place each comp into specified position by layout setup
- Components
  - response for user input and output
  - input username, password
  - press button (yes or no)
  - select options (date, male, level)

Container

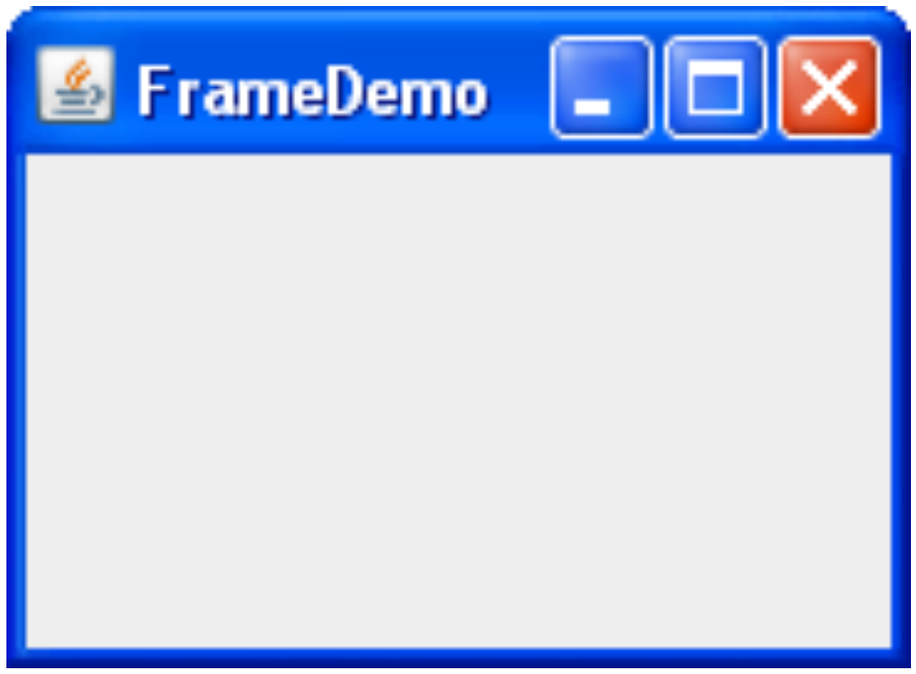
- Top Container
  - connected with OS
    - are not contained in anything else
- intermediate Container
  - manage component
  - can be nested

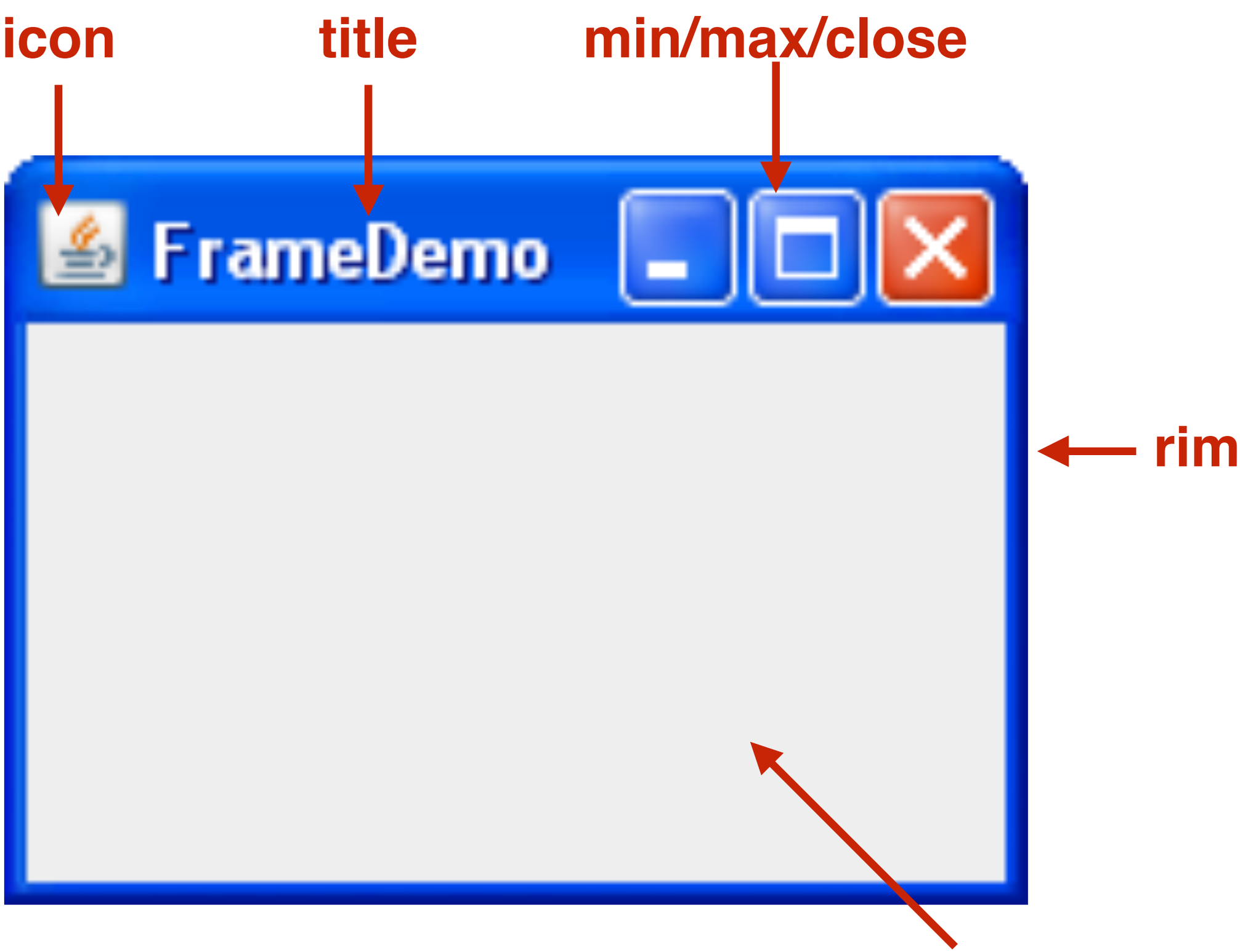
Top Container

- Top Container
  - JFrame
    - usually we will use this
  - JApplet
    - used in browser
  - JDialog
    - for simple use
  - notfiy the **J**
    - all the swing classes begin with **J**

Container

- To Create a Window in Swing Program
- Including the Rim, Title, Icon, Min/Max/Close and ContentPane
- Constructor
  - JFrame()
  - JFrame(String title)




icon
title
min/max/close
FrameDemo
rim
ContentPane

Create a JFrame

```
public class HelloWorldSwing {
    private static void createAndShowGUI() {
        //Create and set up the window.
        JFrame frame = new JFrame("HelloWorldSwing");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        //Add the ubiquitous "Hello World" label.
        JLabel label = new JLabel("Hello World");
        frame.getContentPane().add(label);

        //Display the window.
        frame.setVisible(true);
    }
    public static void main(String[] args) {
        createAndShowGUI();
    }
```

intermediate Container

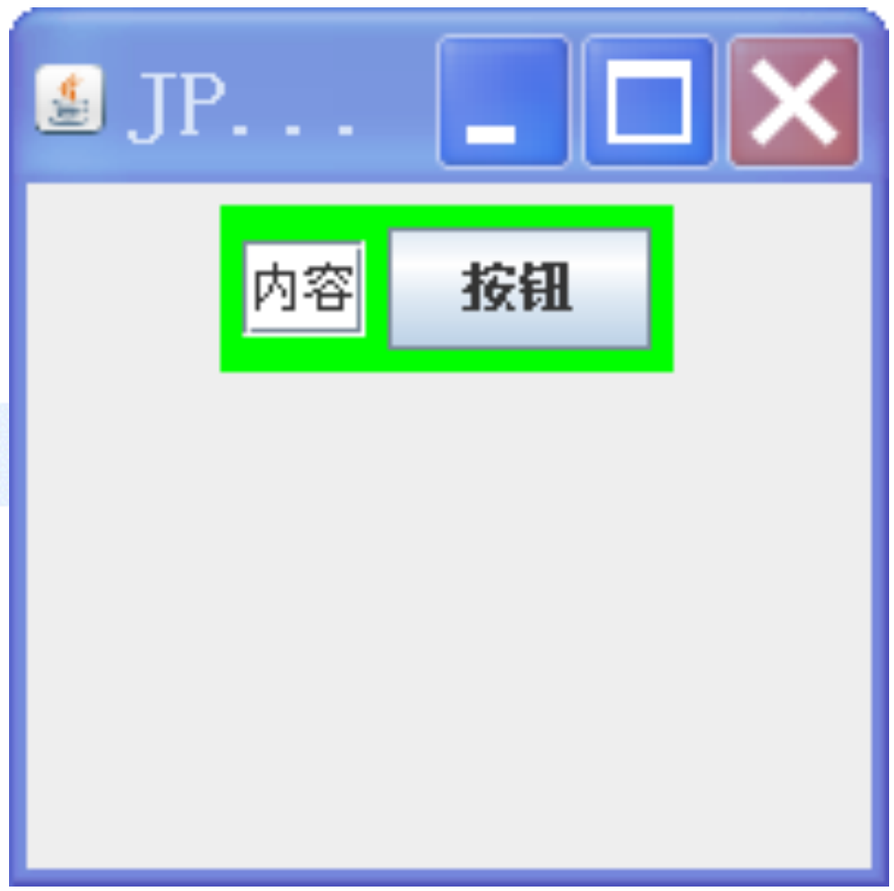
- intermediate Container
  - JPanel
  - JSplitPane
  - JScrollPane

JPanel

- Middle-level Container
- Combining Small Light-weighted Component
- Constructor
  - JPanel()
  - JPanel(boolean isDoubleBuffered)
  - JPanel(LayoutManager layout)
  - JPanel(LayoutManager layout, boolean isDoubleBuffered)

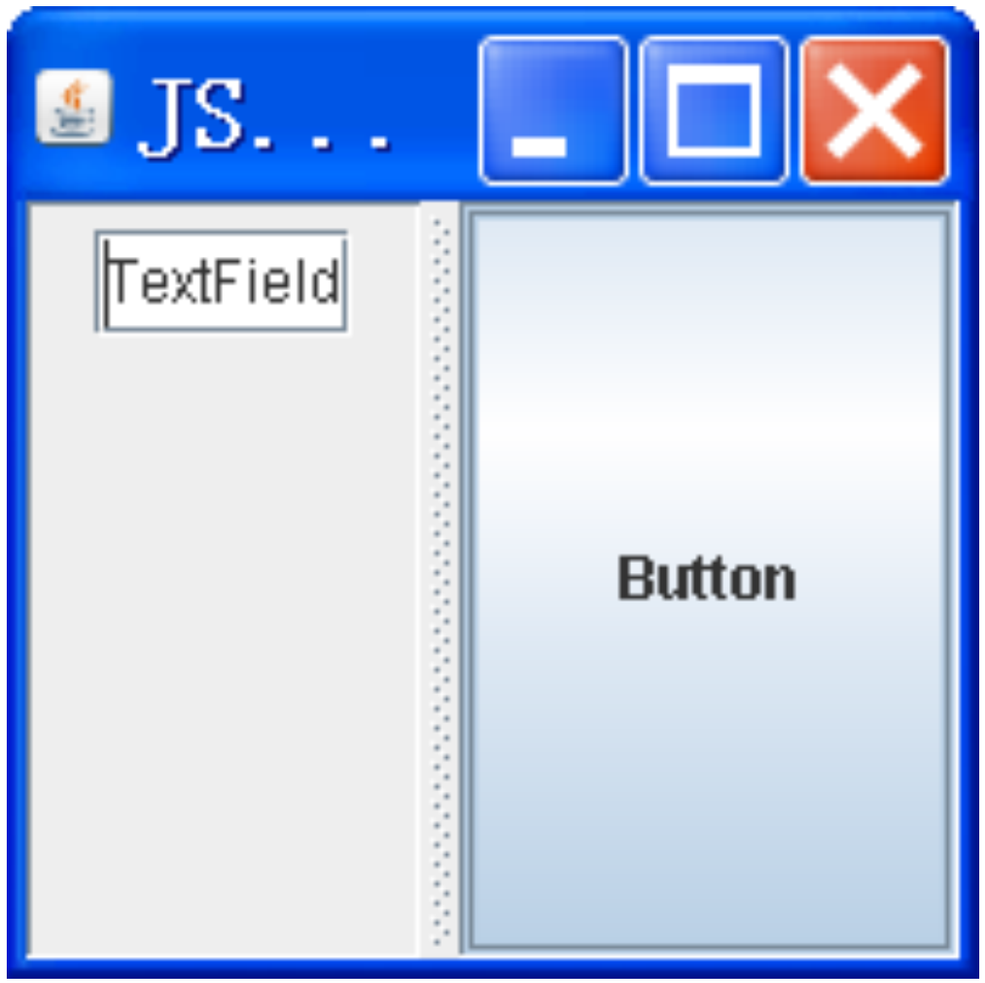
JPanel

```
JFrame f = new JFrame("JPanel example");
f.setDefaultCloseOperation(EXIT_ON_CLOSE);
Container cp = f.getContentPane();
JPanel p1 = new JPanel();
p1.setBackground(Color.green);
p1.add(new JTextField("内容"));
p1.add(new JButton("按钮"));
f.getContentPane().add(p1);
f.setSize(200, 200);
f.setVisible(true);
```



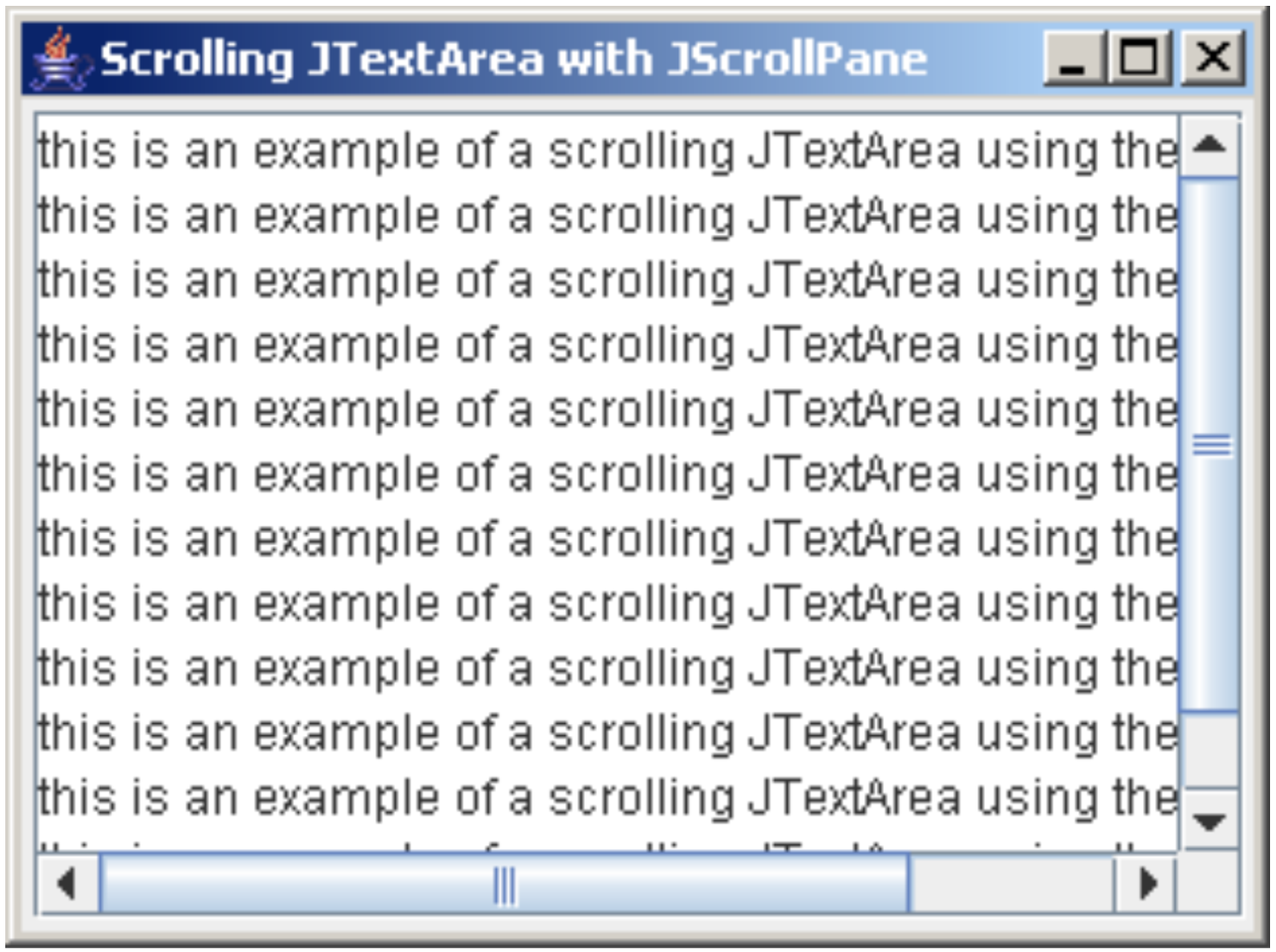
Other Panels

- JSplitPane

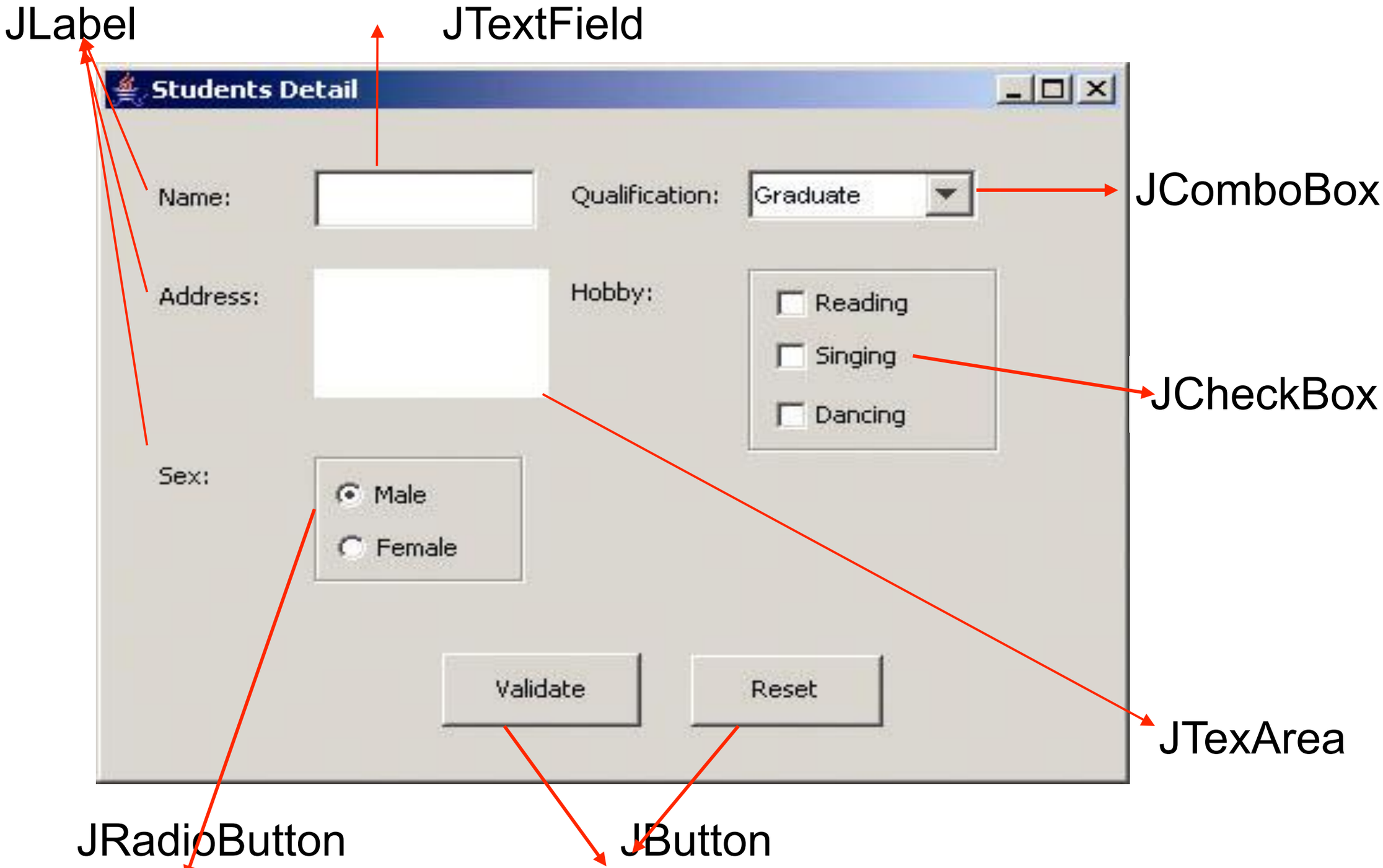


Other Panels

- JScrollPane



Components


JLabel
JTextField
Students Detail
Name:
Qualification:
Graduate
JComboBox
Address:
Hobby:
Reading
Singing
Dancing
JCheckBox
Sex:
Male
Female
Validate
Reset
JTexArea
JRadioButton
JButton

TextComponent

- JLabel
- JTextField
- JTextArea
- JPasswordField

JLabel

- Constructor
  - JLabel()
  - JLabel(String text)
  - JLabel(Icon image)
- Methods
  - String getText()、 void setText(String text)
  - void setIcon(Icon icon)

JTextField

- Constructor

  - JTextField()
  - JTextField(String text)

- Methods

```
boolean isEditable(); void setEditable(boolean b)
int getColumns(); void setColumns(int columns)
int getHorizontalAlignment; void setHorizontalAlignment(int value)
String getSelectedText()
void setSelectionEnd(int selectionEnd)
void setSelectionStart(int selectionStart)
```

JPasswordField

- get/setEchoChar()
- getPassword()

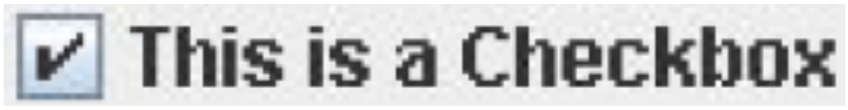
Button

- JButton
- JCheckBox
- JRadioButton

JButton

- Constructor
  - JButton(); JButton(Icon icon)
  - JButton(String text); JButton(String text, Icon icon)
- Methods
```
boolean isDefaultButton()
String getText(); void setText(String text)
String getActionCommand()
void setActionCommand(String actionCommand)
public ActionListener[] getActionListeners()
public void addActionListener(ActionListener l)
void removeActionListener(ActionListener l)
```

JCheckBox



- Constructor
  - JCheckBox(Icon icon)、JCheckBox(Icon icon, boolean selected)
  - JCheckBox(String text)、JCheckBox(String text, boolean selected)
  - JCheckBox(String text, Icon icon)、JCheckBox(String text, Icon icon, boolean selected)
- Methods
  - boolean isSelected()、void setSelected(boolean b)
  - public ActionListener[] getActionListeners()、public void addActionListener(ActionListener l)、void removeActionListener(ActionListener l)

JRadioButton

- Constructor
  - JRadioButton(Icon icon)、JRadioButton(Icon icon, boolean selected)
  - JRadioButton(String text)、JRadioButton(String text, boolean selected)
  - JRadioButton(String text, Icon icon)、JRadioButton(String text, Icon icon, boolean selected)
- Methods
  - boolean isSelected()、void setSelected(boolean b)
  - public ActionListener[] getActionListeners()、public void addActionListener(ActionListener l)、void removeActionListener(ActionListener l)

ButtonGroup

- To Group JRadioButton
- Constructor
  - ButtonGroup()
- Methods
  - int getButtonCount()
  - void add(AbstractButton
  - void remove(AbstractButton b)

Other Components

- JList

- JTree

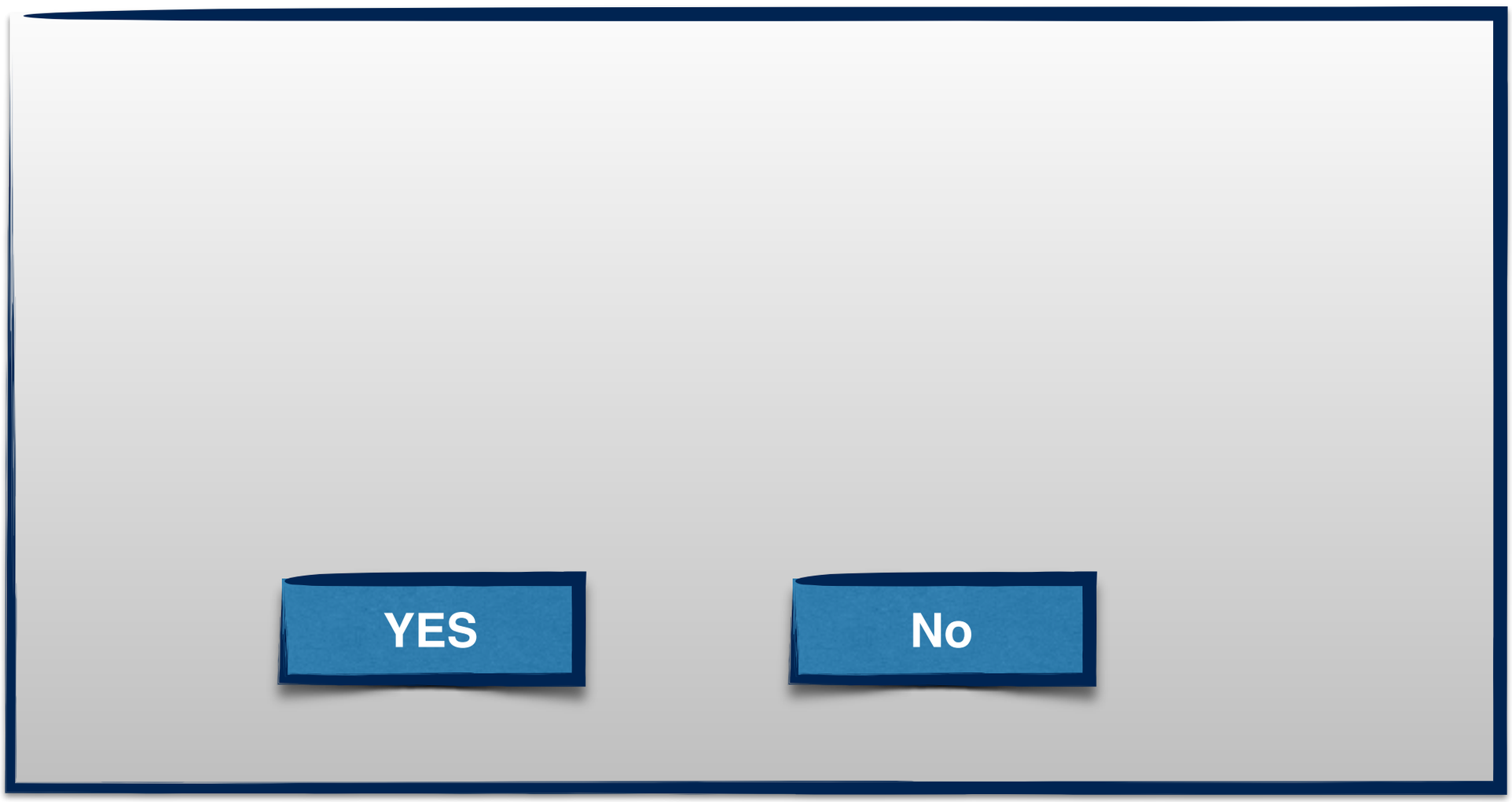
A	B	C	D	E
0x0	0x1	0x2	0x3	0x4
1x0	1x1	1x2	1x3	1x4
2x0	2x1	2x2	2x3	2x4
3x0	3x1	3x2	3x3	3x4
4x0	4x1	4x2	4x3	4x4

Layout Manager

Layout Manage

- Why we need Layout Manage?
  - Portability
  - Dynamic Layout

if we resize the window, where is the button?



the button is outside of the window?

YES

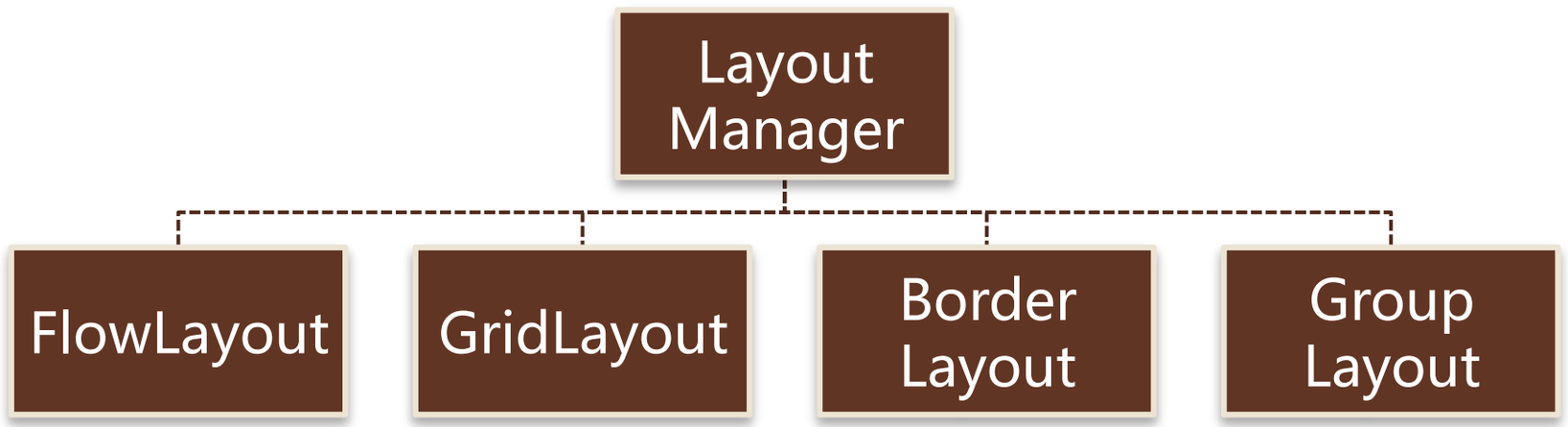
No

Layout Manage

- Function
  - Assemble the components in an ordered way
    - Set the size, position of components
    - Know how to adapt when frame is moved or resized
- Different LM Uses Different Algorithm and Policy
- Each Container has a default Layout Manager

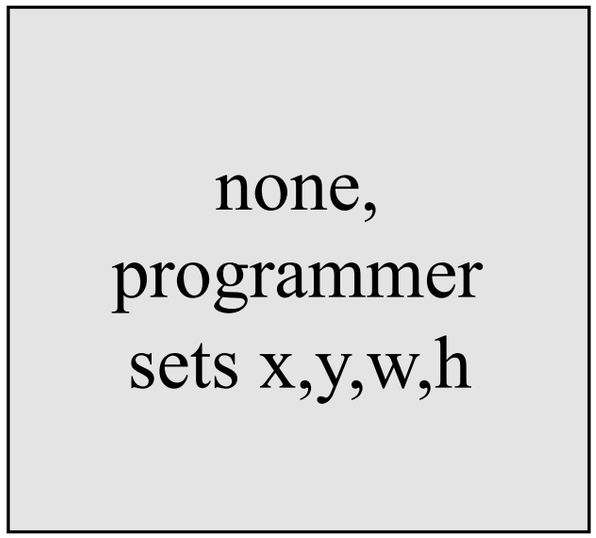
Layout Manager

- Only Container and Subclasses Can Set Layout
- Setting Layout: setLayout(new xxxLayout())
- Common Layout Manager
  - FlowLayout
  - BorderLayout
  - GridLayout
  - GroupLayout

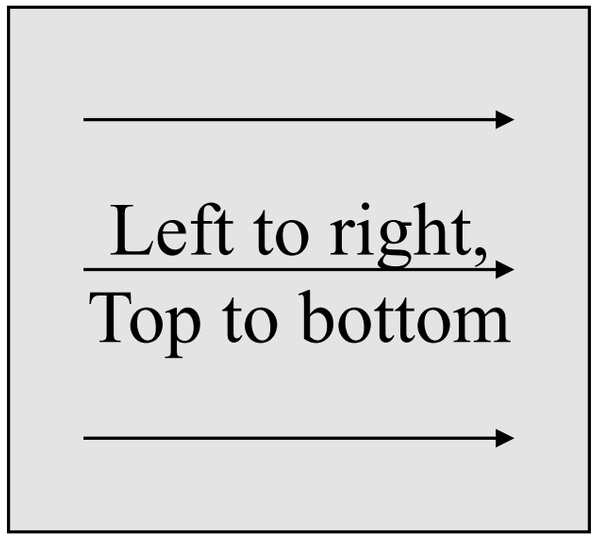


Layout Manager Heuristics

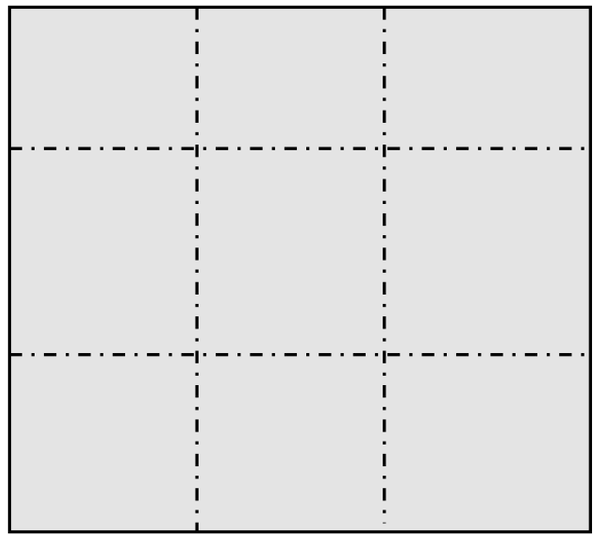
null



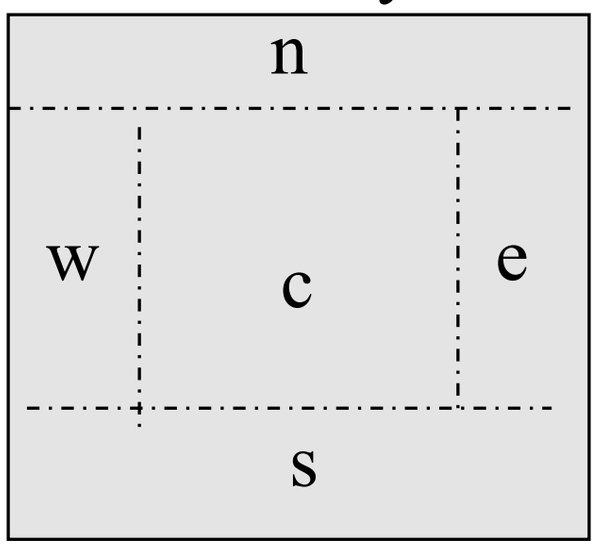
FlowLayout



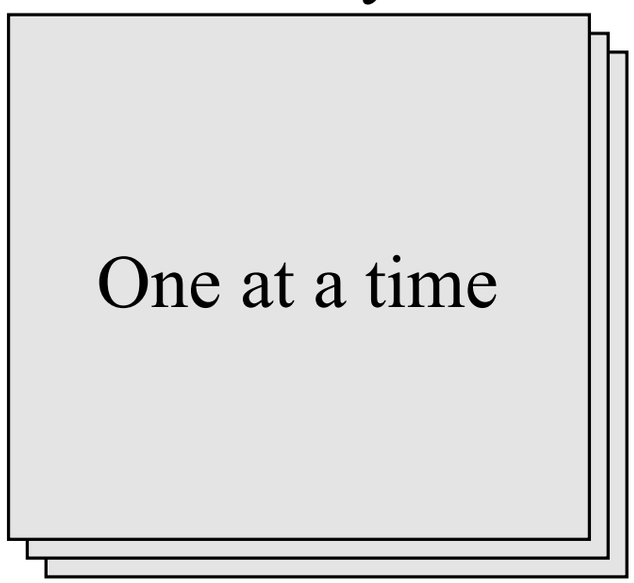
GridLayout



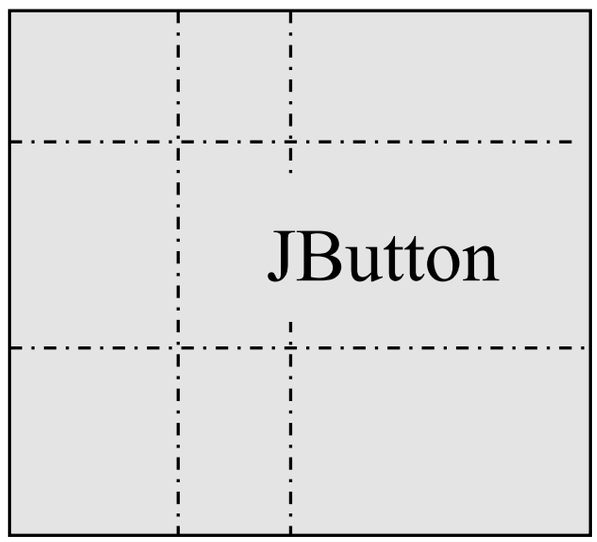
BorderLayout



CardLayout

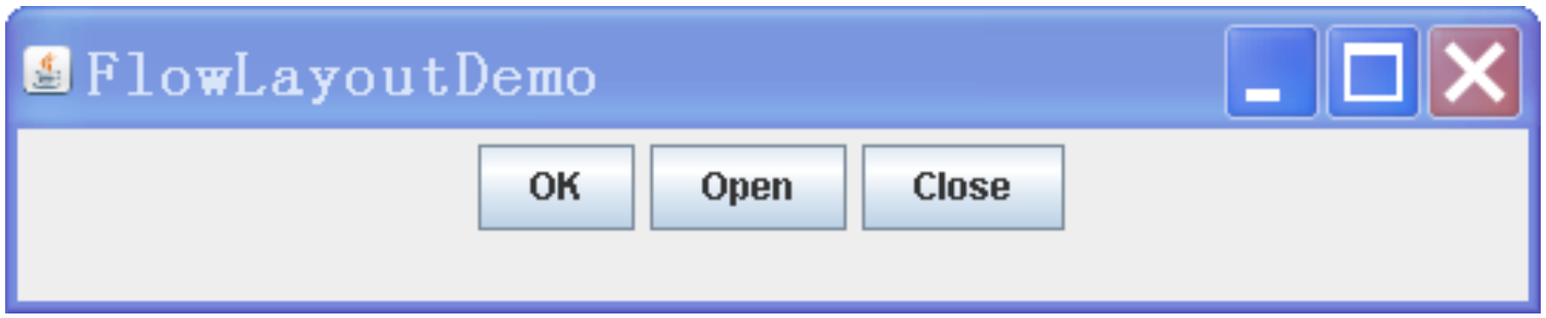


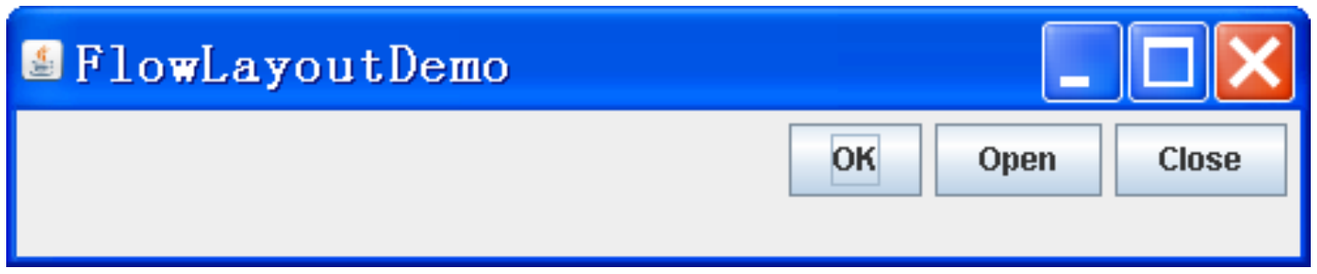
GridBagLayout



Flow Layout

- Default LM for JPanel
- Constructor
  - FlowLayout()
  - FlowLayout(int align)
  - FlowLayout(int align, int hgap, int vgap)
- Methods
  - int getAlignment()、void setAlignment(int align)
  - int getHgap()、void setHgap(int hgap)
  - int getVgap()、void setVgap(int vgap)



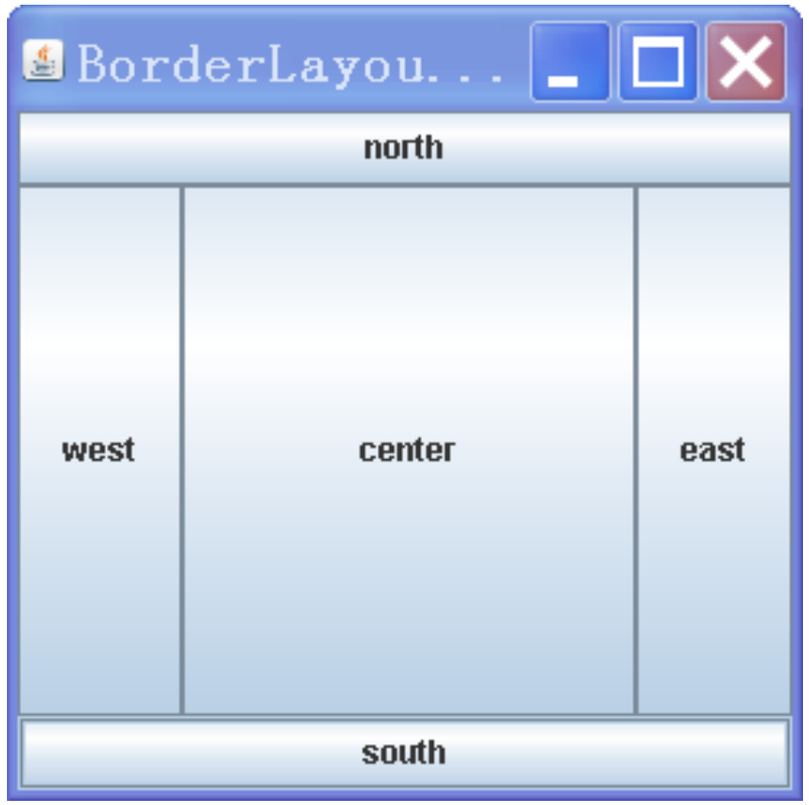



```
JFrame frame = new JFrame("FlowLayoutDemo");
frame.getContentPane().setLayout(
        new FlowLayout(FlowLayout.RIGHT));
JButton button1 = new JButton("OK");
JButton button2 = new JButton("Open");
JButton button3 = new JButton("Close");
frame.add(button1);
frame.add(button2);
frame.add(button3);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setSize(500, 100);
frame.setVisible(true);
```

Border Layout

- Divide the Container into 5 Zones: North/South/East/West/Center
- Default LM for JFrame
- Constructor
  - BorderLayout()
  - BorderLayout(int hgap, int vgap)
- Methods
  - int getAlignment()、void setAlignment(int align)
  - int getHgap()、void setHgap(int hgap)
  - int getVgap()、void setVgap(int vgap)

```
JFrame frame = new JFrame("BorderLayoutDemo");
frame.getContentPane().setLayout(
        new BorderLayout());
JButton button1 = new JButton("center");
JButton button2 = new JButton("east");
...
frame.add(button1, BorderLayout.CENTER);
frame.add(button2,BorderLayout.EAST);
...
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setSize(300, 300);
frame.setVisible(true);
```



Reference

- The Swing Tutorial : http://docs.oracle.com/javase/tutorial/uiswing/index.html

- SwingSet : http://swingset.sourceforge.net/