Fpt University

FPT POLYTECHNIC

LẬP TRÌNH JAVA 3

BÀI 1: AWT, SWING, CONTAINER

PHẦN 1

www.poly.edu.vn

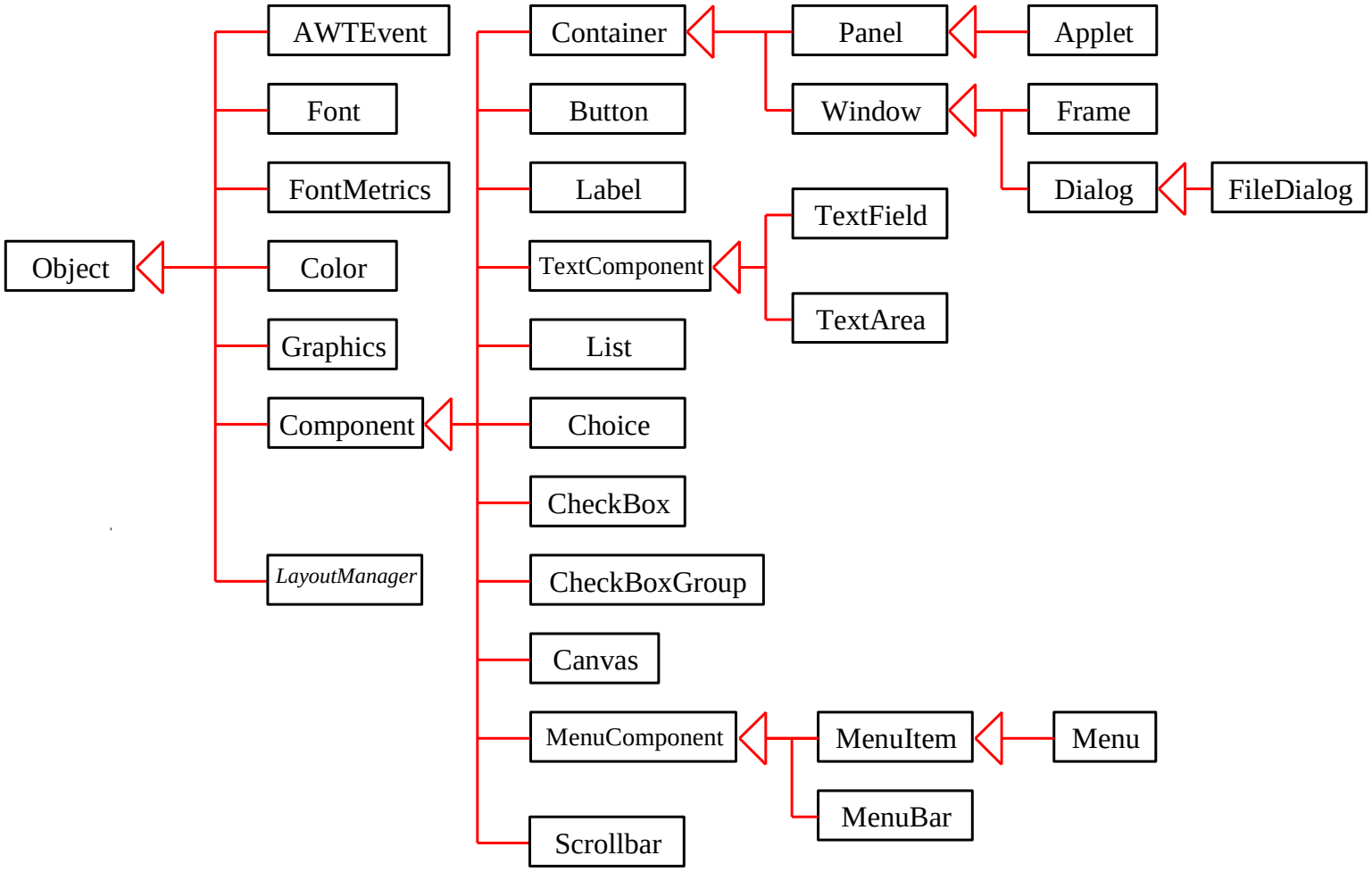
Mục tiêu

- Kết thúc bài học này bạn có khả năng
  - Giới thiệu gói thư viện AWT
  - Giới thiệu SWING
  - So sánh AWT và Swing
  - Container trong Java Swing
  - Container Component (JFrame, JPanel)
  - Demo

GIỚI THIỆU GÓI THƯ VIỆN AWT

- AWT viết tắt của Abstract Windowing Toolkit
- AWT là tập hợp các lớp Java cho phép chúng ta tạo một GUI.
- Cung cấp các mục khác nhau để tạo hoạt động và hiệu ứng GUI
- Tạo liên kết giao diện giữa ứng dụng Java và OS
- Chiếm nhiều tài nguyên hệ thống (Heavy-weight component)
- Package java.awt
- Gồm nhiều phần tử (class) để tạo GUI.
- Có (event-oriented application) mô hình ứng dụng hướng sự kiện.
- Có các công cụ xử lý đồ họa và hình ảnh.

GIỚI THIỆU GÓI THƯ VIỆN AWT

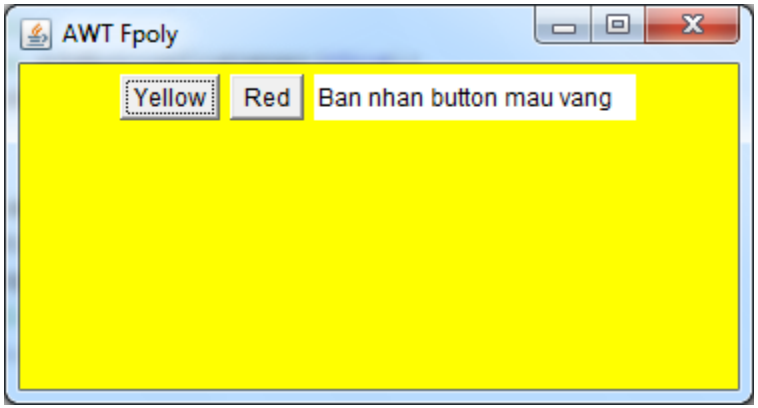


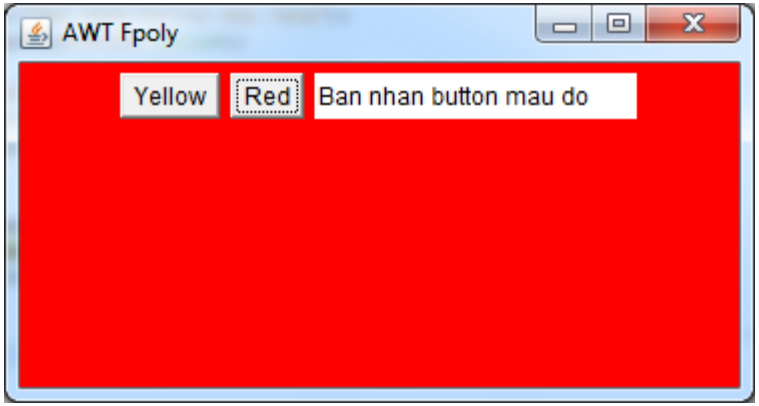
GIỚI THIỆU GÓI THƯ VIỆN AWT

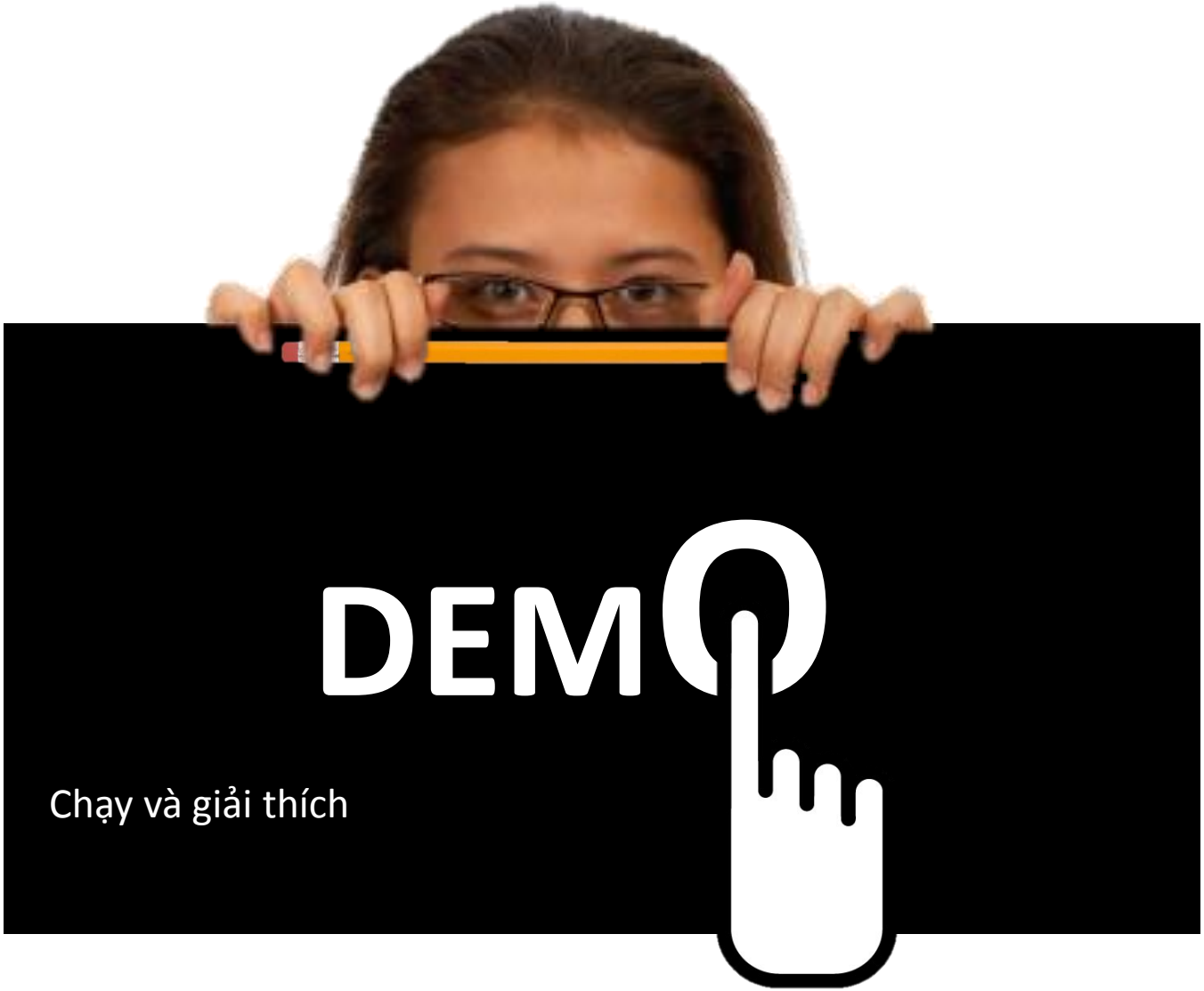
- Các nguyên tắc xây dựng ứng dụng GUI
  - Lựa chọn một container: Frame, Window, Dialog, Applet,…
  - Tạo các control: (buttons, text areas, list, choice, checkbox,…)
  - Đưa các control vào vùng chứa
  - Sắp xếp các control trong vùng chứa (Layout).
  - Thêm các xử lý sự kiện (Listeners)

GIỚI THIỆU GÓI THƯ VIỆN AWT

```
public class Demo1AWT extends Frame implements ActionListener{
    Button btnYellow,btnRed;Label label = new Label();
    public Demo1AWT(String msg){
        setTitle(msg);
        setLayout(new FlowLayout());
        btnYellow = new Button("Yellow");
        btnRed = new Button("Red");
        add(btnYellow);
        add(btnRed);
        add(label);
        btnYellow.addActionListener(this);
        btnRed.addActionListener(this);
    }
    @Override
    public void actionPerformed(ActionEvent e) {
        String str = e.getActionCommand();
        if(str.equals("Yellow")){
            label.setText("Ban nhan button mau vang");
            this.setBackground(Color.yellow);
        }
        if(str.equals("Red")){
            label.setText("Ban nhan button mau do");
            this.setBackground(Color.RED);
        }
    }
    public static void main(String[] args) {
        Demo1AWT ab = new Demo1AWT("AWT Fpoly");
        ab.setSize(450, 200);
        ab.show();
    }
}
```







Chạy và giải thích

GIỚI THIỆU SWING

- SWING là sự mở rộng AWT
- Tạo giao diện nhất quán độc lập với môi trường
- Đa luồng và nhẹ(Light-weight component) với nhiều đặc điểm nâng cấp
- Có khả năng tùy biến tại thời điểm runtime
- Không sử dụng trộn lẫn AWT và SWING GUI component trên cùng một giao diện
- Package java.swing
- Sự xuất hiện thêm Swing từ Java 1.2 nhằm giúp khắc phục sự khó khăn của AWT khi cần bổ sung thêm các widget mới
- Đặc biệt, khắc phục bộ mặt nghèo nàn của chương trình viết bằng AWT so với các giao diện khác (chẳng hạn với MFC (Microsoft Foundation Classes) của Microssoft

So sánh AWT và SWING

AWT

- Xây dựng bằng native code
- Khó phát triển thêm các linh kiện(widget) mới

SWING

- Xây dựng hoàn toàn bằng JAVA API
- Dễ phát triển các linh kiện
- Có thể thay đổi diện mạo của linh kiện lúc runtime
- Mô hình MVC (Model – View – Controller)

So sánh AWT và SWING

AWT	SWING
	Có thể bổ sung thêm biểu tượng bên cạnh dòng chữ vào Label, Button, Menu item ▪ Tạo đường viền và ghi tựa cho các thành phần Swing

So sánh AWT và SWING

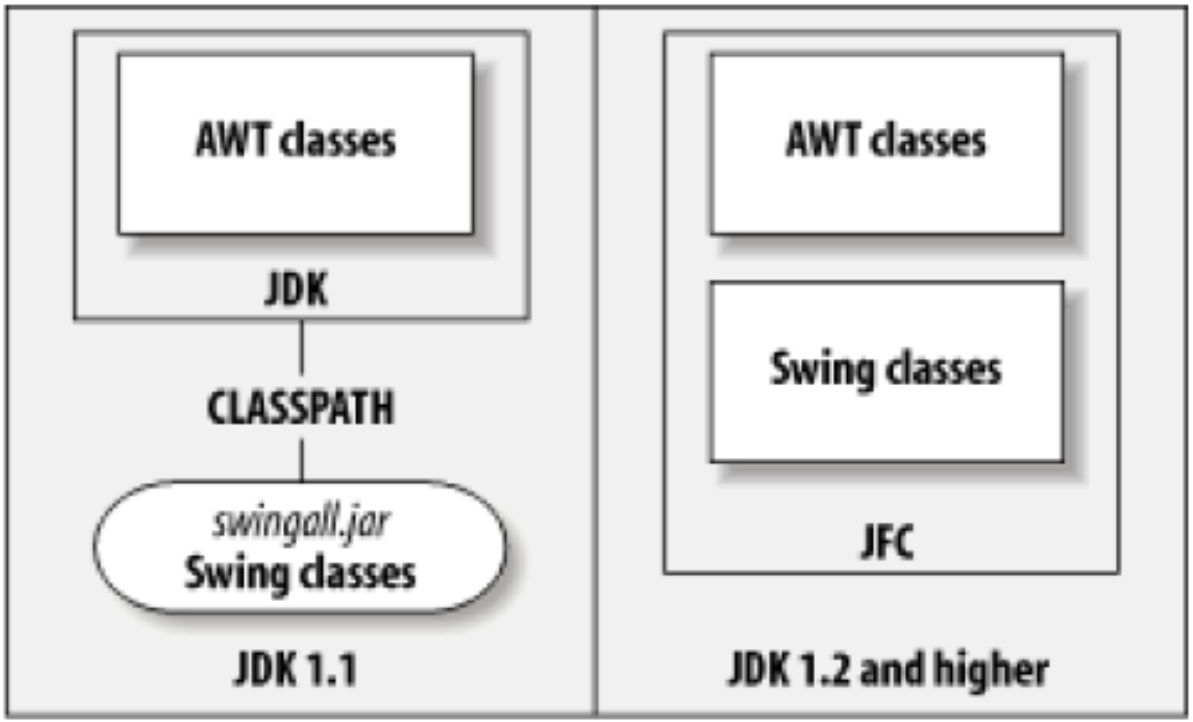
AWT	SWING
	▪ Có thể chọn một thành phần Swing bằng Cách dùng phím thay cho chuột ▪ Các thành phần Swing sử dụng các nhãn Unicode, vì vậy có thể đưa tiếng Việt vào các thành phần này

So sánh AWT và SWING

- Vậy SWING là sự thay thế của AWT?

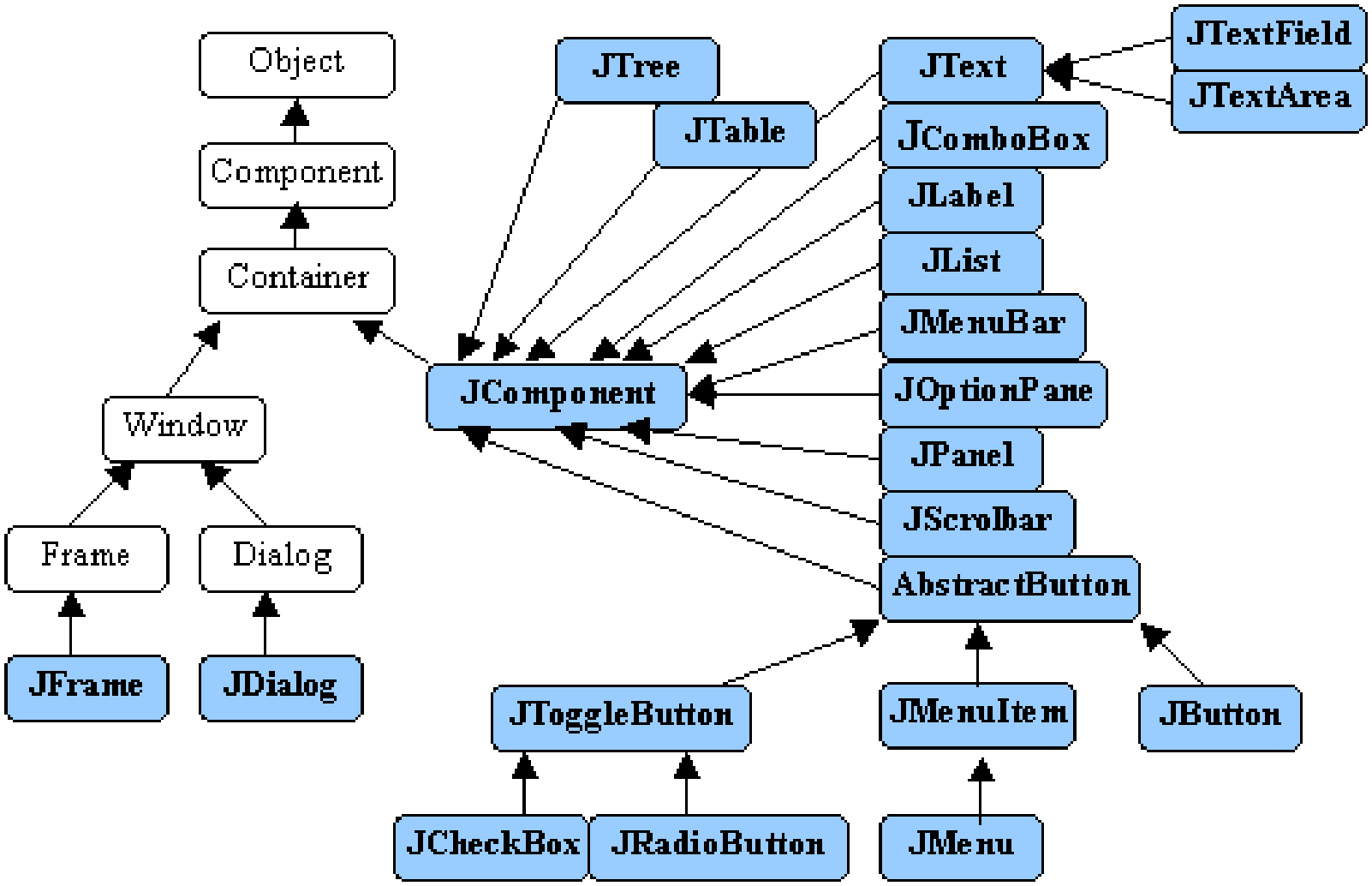
Không. Swing thực tế được xây dựng trên phần lõi của AWT, bởi vì Swing không chứa bất kỳ mã dành cho nền tảng nào (native code)

Mô tả mối quan hệ giữa AWT, SWING, và JDK:



So sánh AWT và SWING

Kiến trúc SWING



CONTAINER TRONG JAVA SWING

- Thành phần chứa trong Swing, hay còn gọi là Container
- Có 2 kiểu Container trong Swing, đó là Top-level Container và Multi-purpose Container
- Swing cung cấp các loại Top-level Container đó là:
  - JFrame
  - JDialog
  - JApplet: được sử dụng cho ứng dụng web
  - JWindow: loại này không có đặc điểm gì cả, chỉ là một màn hình chờ được bật lên trong lúc khởi động (Splash – Screen)
- General-purpose thì gồm có: JPanel, JLayered, JInternalFrame, và JDesktopPane

Khi thêm các thành phần (component) vào JFrame chúng ta không thêm trực tiếp, thay vào đó, phần lớn các thành phần sẽ thêm vào ContentPane bằng cách gọi phương thức: getContentPane().add(component);

Tạo JFrame tự code:

```
1   Import javax.swing.JFrame;
2   public class DemoJframe extends JFrame {
3       // TO DO code here
4       public DemoJframe(){
5       ……
6       ……
7       }
8       public static void main (String [] args){
9       DemoJframe objFrame = new DemoJframe();
10      ……
11      ……
12      }
13  }
14
```

JFRAME

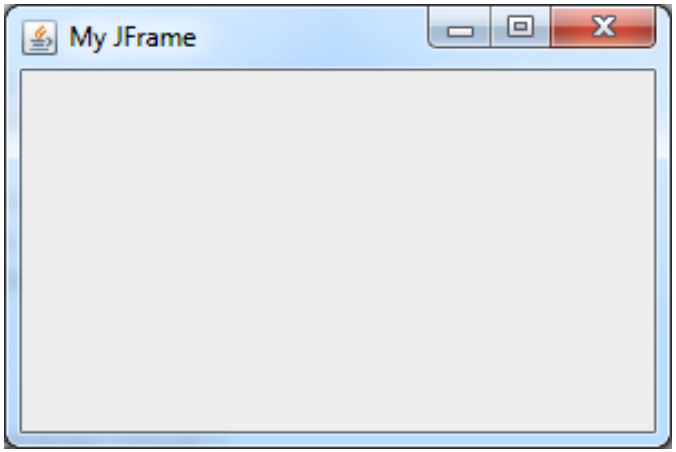
- JFrame sau khi được tạo thì chúng ta sẽ không nhìn thấy được nó. Để nhìn được nó chúng ta phải nhờ đến phương thức
  - **setVisible(boolean value)**.
- Thiết lập độ rộng và cao của JFrame.
  - **setSize(width, heigh)**
- Khi **JFrame** đã được nhìn thấy, muốn đóng lại dùng phương thức
  - **setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);**
  - Nếu không dùng hàm này để đặt, thì mặc định là: HIDE_ON_CLOSE: Khi đóng frame sẽ bị ẩn đi chứ hoàn toàn không đóng lại.
  - Các lựa chọn khác gồm:
    - ✓ DO_NOTHING_ON_CLOSE (0) – không làm gì cả
    - ✓ DISPOSE_ON_CLOSE (2) – Chỉ đóng frame đó, các frame khác liên quan sẽ không bị đóng.
    - ✓ EXIT_ON_CLOSE (3) – Đóng toàn bộ các frame liên quan tới nó.

JFRAME

- Đặt vị trí xuất hiện của JFrame trên màn hình.
  - **setLocation(x,y)**
- Đặt JFrame xuất hiện chính giữa màn hình
  - **setLocationRelativeTo(null)**
- Đặt kích thước Jframe vừa đủ với nội dung
  - **pack()**
- Đặt JFrame có thể được thay đổi kích thước hay không,mặc định là True.
  - **setResizable(boolean)**
- Đặt màu nền cho JFrame
  - **getContentPane().setBackground(Color.”Color”)**

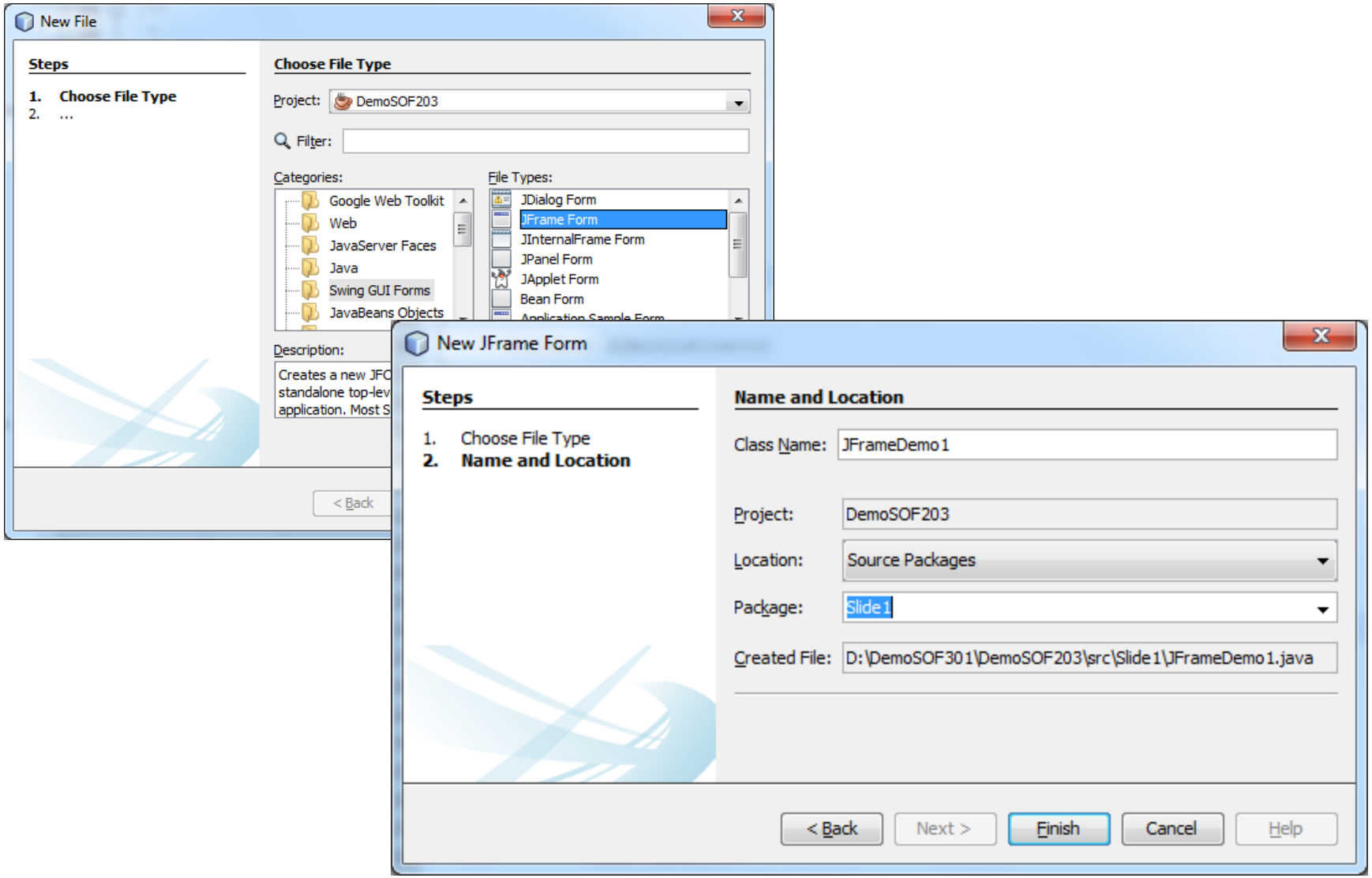
JFRAME-VÍ DỤ CODE

```
public class Demo2Swing extends JFrame {
    public static void main(String[] args) {
        JFrame frame = new JFrame("My JFrame");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 200);
        frame.setVisible(true);
    }
}
```



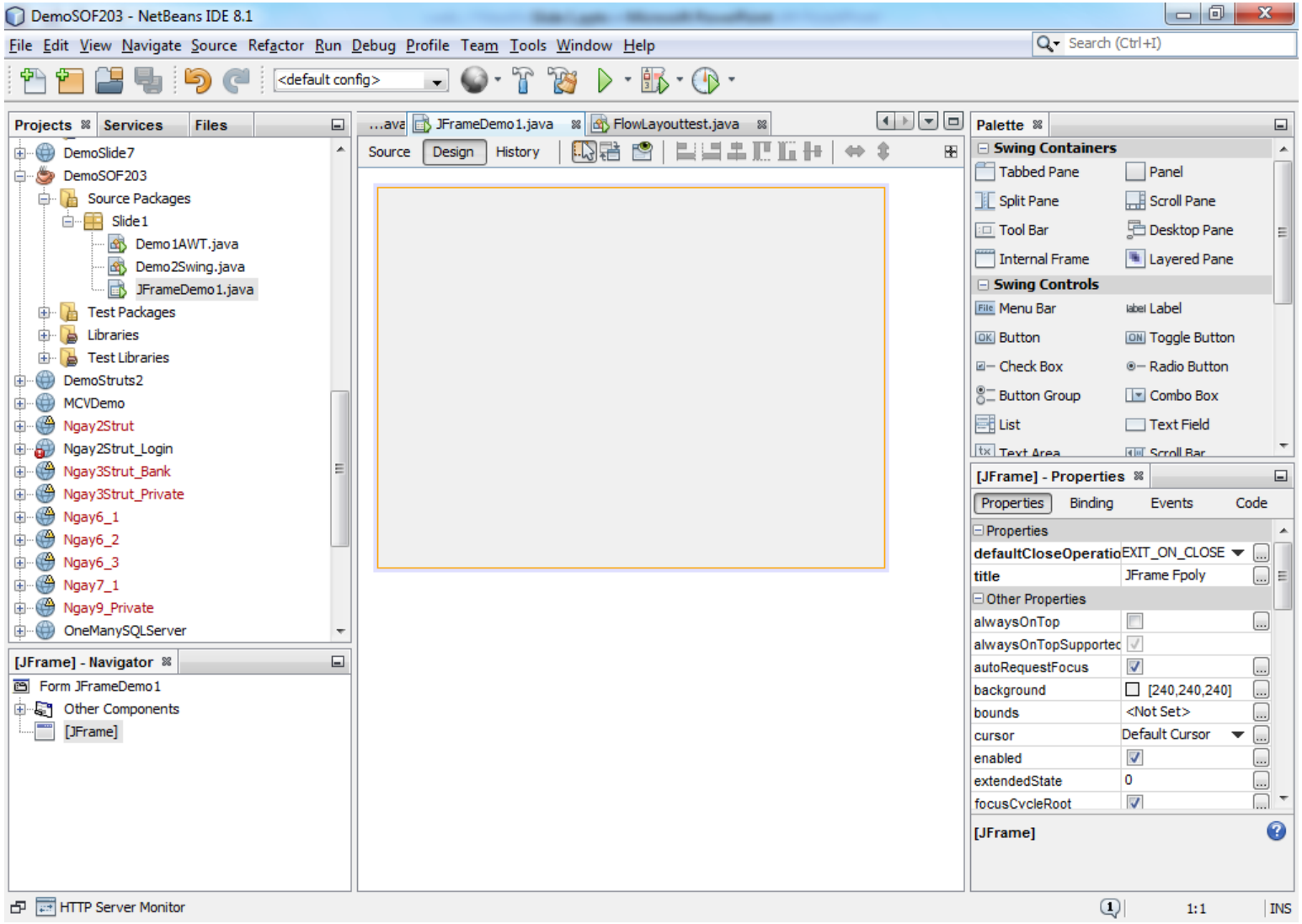
JFRAME-VÍ DỤ NETBEANS

- Dùng Netbean



JFRAME-VÍ DỤ NETBEANS

- Dùng Netbean

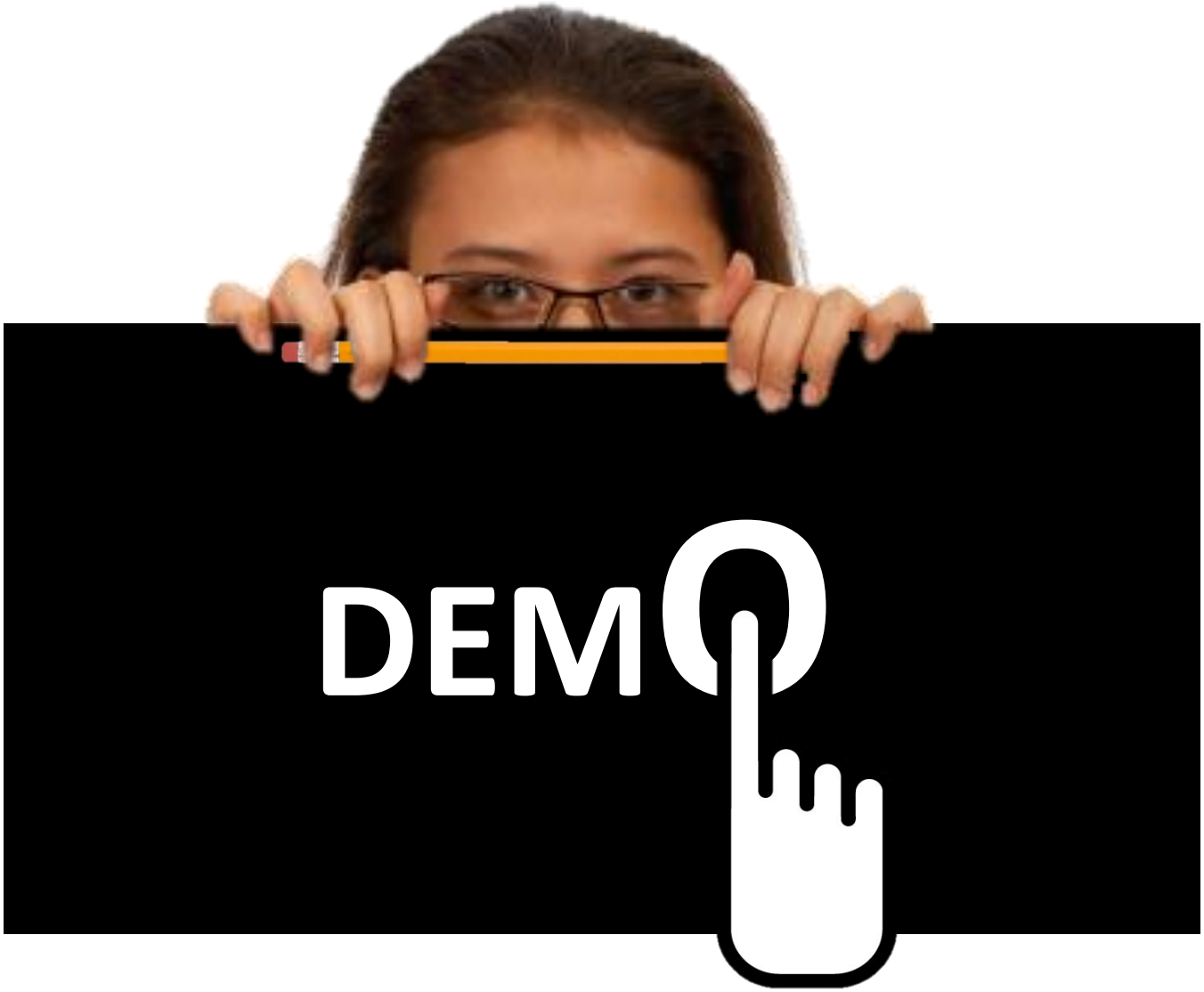


JFRAME-VÍ DỤ NETBEANS

- Dùng Netbean

```
public class JFrameDemo1 extends javax.swing.JFrame {

    /** Creates new form JFrameDemo1 ...3 lines */
    public JFrameDemo1() {
        initComponents();
    }

    /** This method is called from within the cons
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="
    private void initComponents() {...18 lines }//

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        Look and feel setting code (optional)

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new JFrameDemo1().setVisible(true);
            }
        });
    }
```

Code generated


DEMO

Fpt University

FPT POLYTECHNIC

LẬP TRÌNH JAVA 3

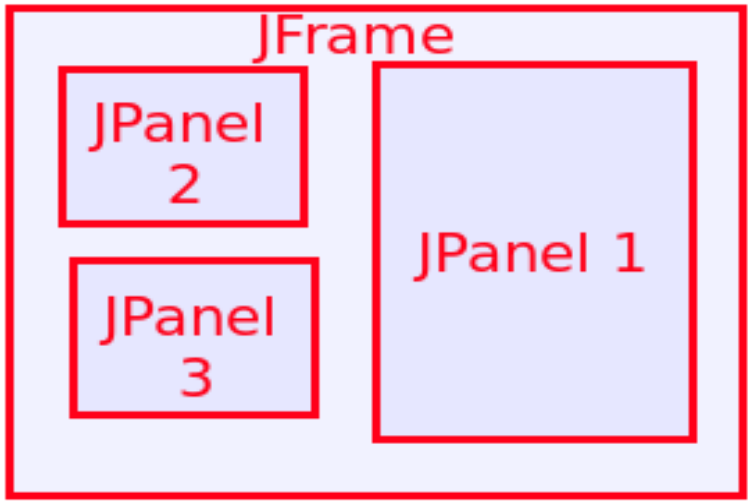
BÀI 1: AWT, SWING, CONTAINER

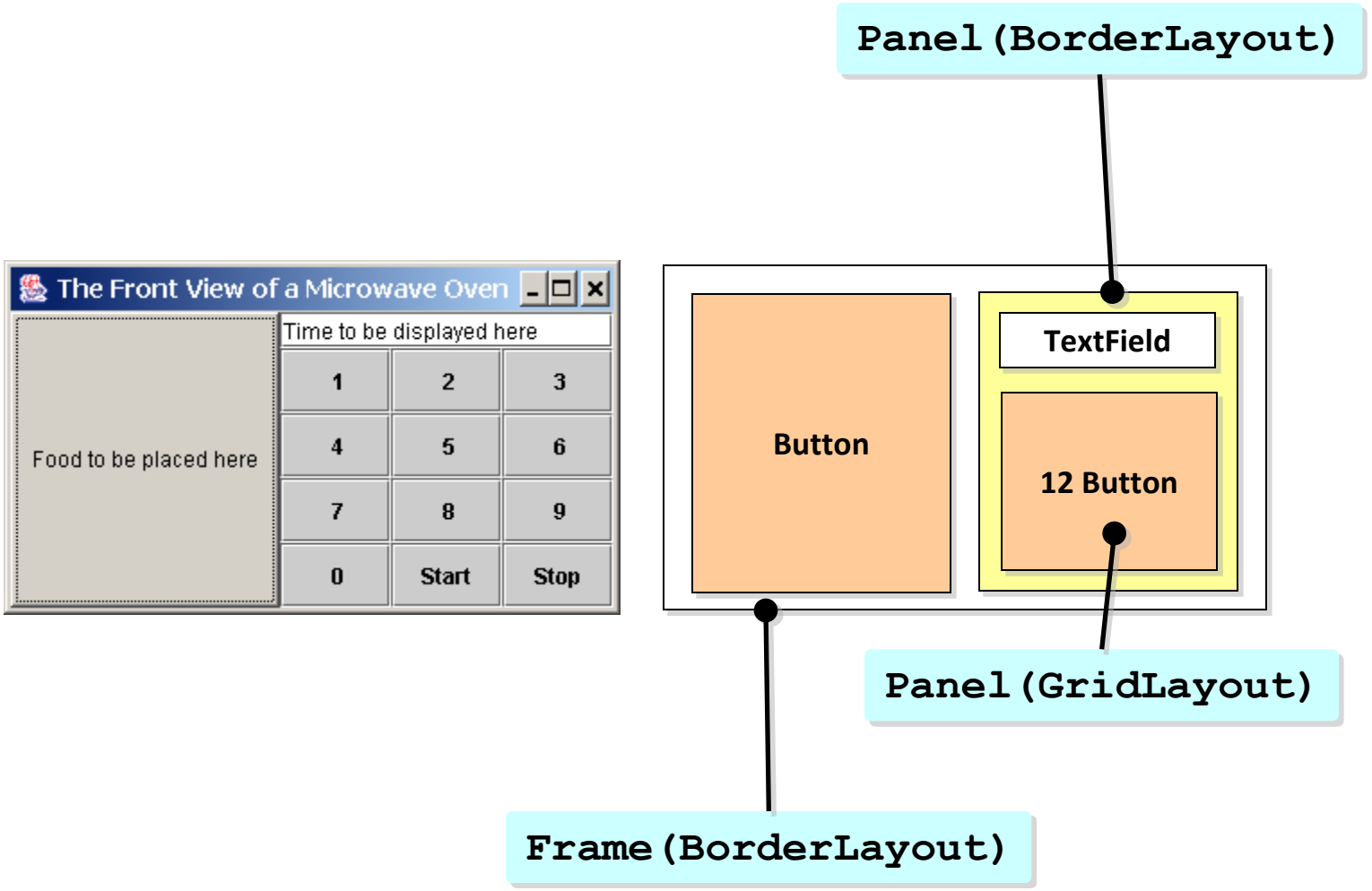
PHẦN 2

www.poly.edu.vn

JPANEL

- JPanel là một container dùng để chứa các thành phần đồ họa khác (tương tự như JFrame tuy nhiên nó không phải là 1 JFrame).
- Trong một JFrame chứa các JPanel, trong mỗi JPanel lại có thể chứa các đối tượng hoặc thậm chí là các JPanel khác.
- Ví dụ: Một giao diện có thể có nhiều panel sắp xếp theo một layout nhất định, mỗi panel lại có các component sắp xếp theo một layout riêng.
- Panel có bố cục mặc định là FlowLayout.




Panel (BorderLayout)
The Front View of a Microwave Oven
Time to be displayed here
Food to be placed here
1
2
3
4
5
6
7
8
9
0
Start
Stop
Button
TextField
12 Button
Panel (GridLayout)
Frame (BorderLayout)

JPANEL

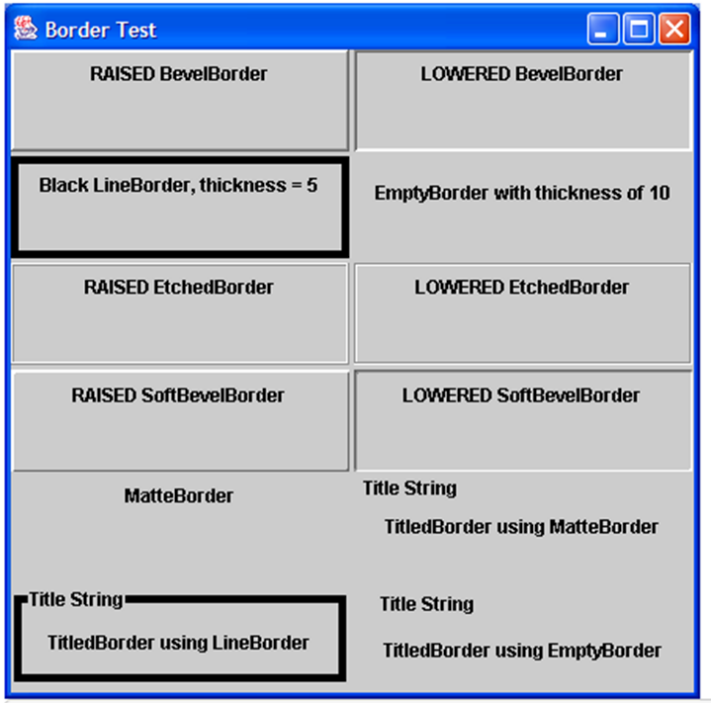
- Chúng ta có 2 Phương thức khởi tạo JPanel đó là:
  - JPanel(): Tạo 1 JPanel với Layout **mặc định** là FlowLayout(cách bố trí mà các đối tượng nối tiếp nhau).
  - JPanel(LayoutManager layout): Tạo 1 JPanel với Layout được **chỉ định**.
- Ví dụ:

```
// create panel1 with layout default is FlowLayout
JPanel panel1 = new JPanel();


// create panel2 with GridLayout
JPanel panel2 = new JPanel(new GridLayout());
```

JPANEL

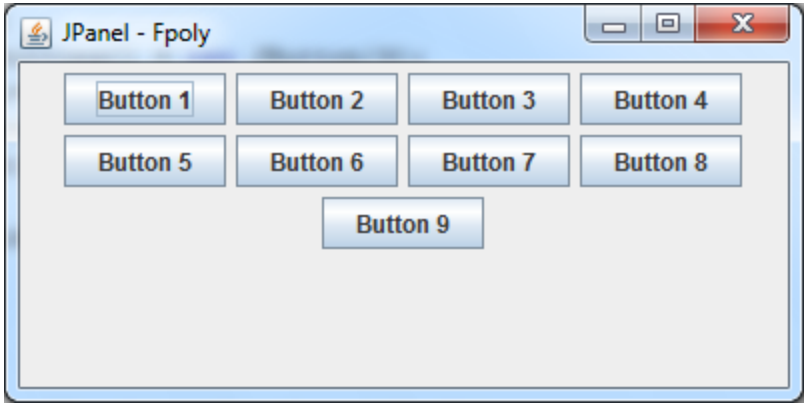
- Đặt Layout cho JPanel
  - **setLayout("layout")**
- Thiết lập vị trí và kích thước cho JPanel
  - **setBound(x,y,width,heigh)**
- Đặt màu nền cho JPanel
  - **setBackground(Color."color")**
- Đặt đường viền cho JPanel
  - Có nhiều kiểu đường viền
  - Sử dụng phương thức **setBorder(new "kiểu-border")**
  - Cùng xem ví dụ

```
p.setBorder(new BevelBorder(BevelBorder.RAISED));
p.setBorder(new BevelBorder(BevelBorder.LOWERED));
p.setBorder(new LineBorder(Color.black, 5));
p.setBorder(new EmptyBorder(10, 10, 10, 10));
p.setBorder(new EtchedBorder(EtchedBorder.RAISED));
p.setBorder(new EtchedBorder(EtchedBorder.LOWERED));
p.setBorder(new SoftBevelBorder(SoftBevelBorder.RAISED));
p.setBorder(new SoftBevelBorder(SoftBevelBorder.LOWERED));
p.setBorder(new MatteBorder(new ImageIcon("BALL.GIF")));
p.setBorder(new TitledBorder(new MatteBorder(new ImageIcon("java2sLogo.gif")), "Title String"));
p.setBorder(new TitledBorder(new LineBorder(Color.black, 5),"Title String"));
p.setBorder(new TitledBorder(new EmptyBorder(10, 10, 10, 10),"Title String"));
```



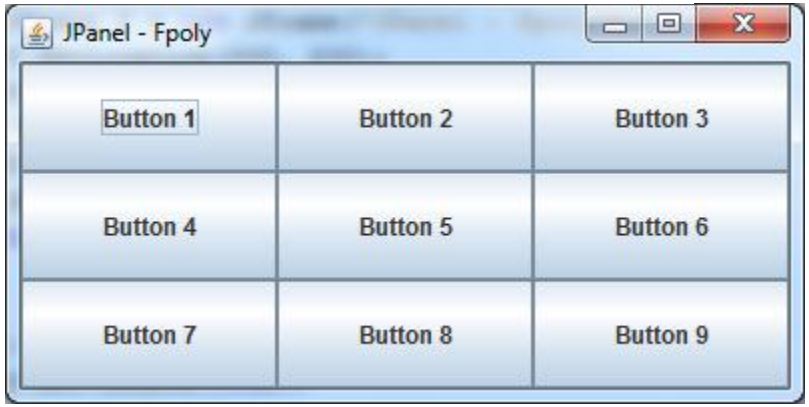
Demo 1:

```
public class JPanelDemo1 {
    public static void main(String[] args) {
        JFrame f = new JFrame("JPanel - Fpoly");
        f.setLocation(200, 200);
        f.setSize(400, 200);
        JPanel p = new JPanel();
        f.add(p);
        JButton buttons[] = new JButton[9];
        for(int i=0;i<9;i++){
            buttons[i]=new JButton("Button "+(i+1));
            p.add(buttons[i]);
        }
        f.setVisible(true);
    }
}
```

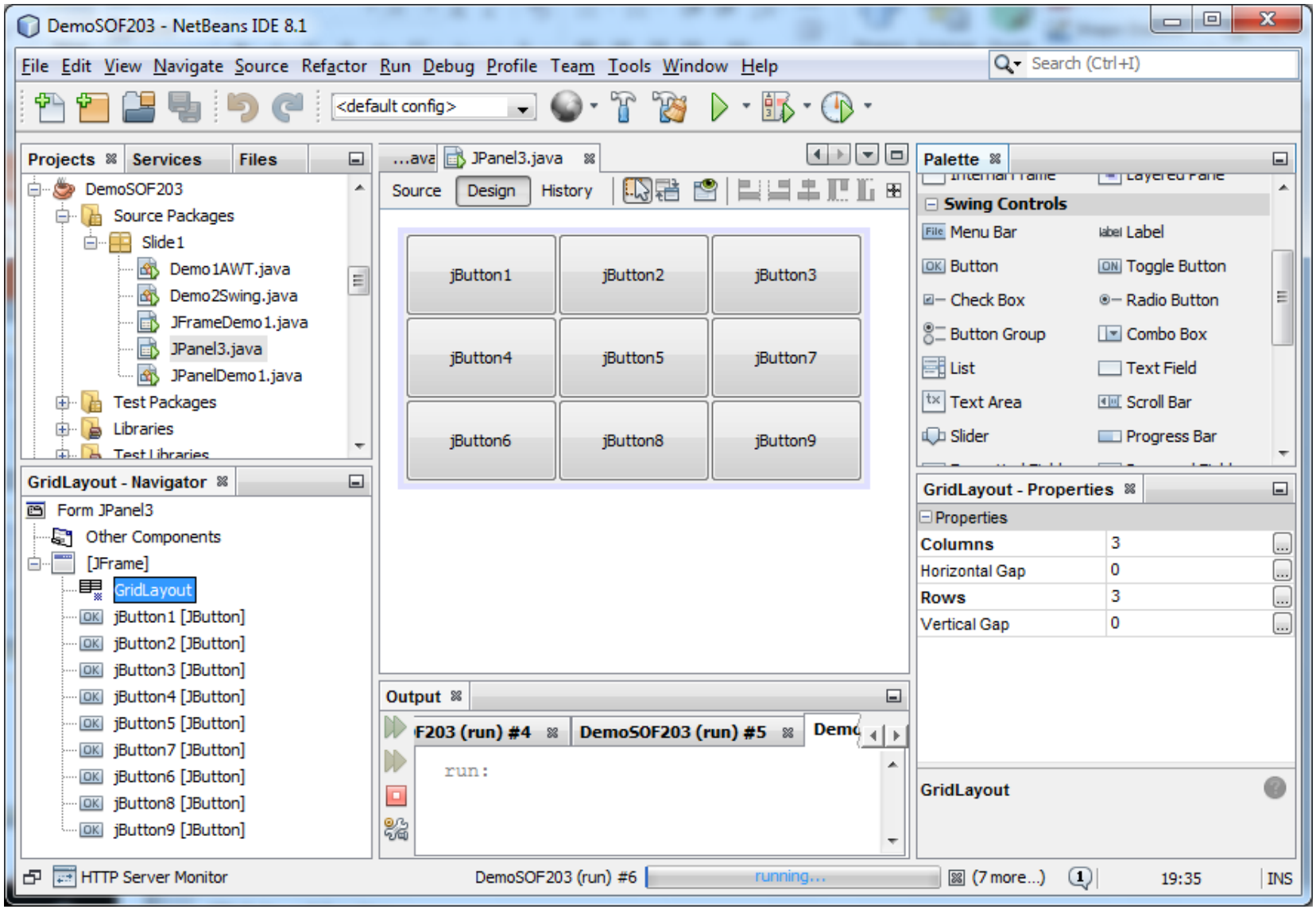


Demo 2:

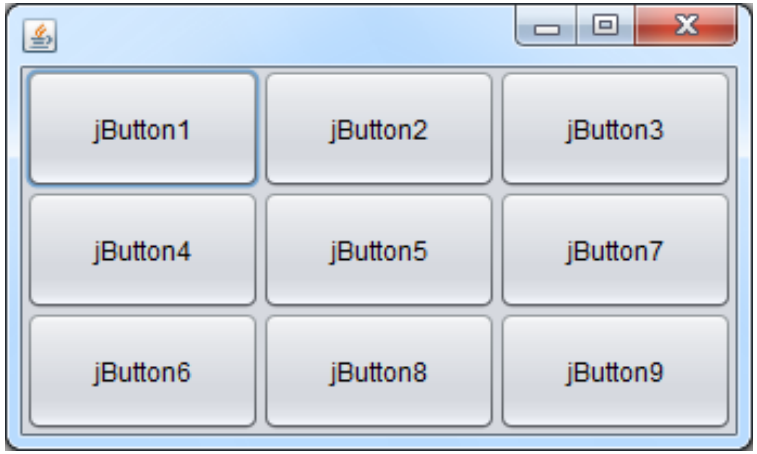
```
public class JPanelDemo1 {
    public static void main(String[] args) {
        JFrame f = new JFrame("JPanel - Fpoly");
        f.setLocation(200, 200);
        f.setSize(400, 200);
        JPanel p = new JPanel(new GridLayout(3, 3));
        f.add(p);
        JButton buttons[] = new JButton[9];
        for(int i=0;i<9;i++){
            buttons[i]=new JButton("Button "+(i+1));
            p.add(buttons[i]);
        }
        f.setVisible(true);
    }
}
```

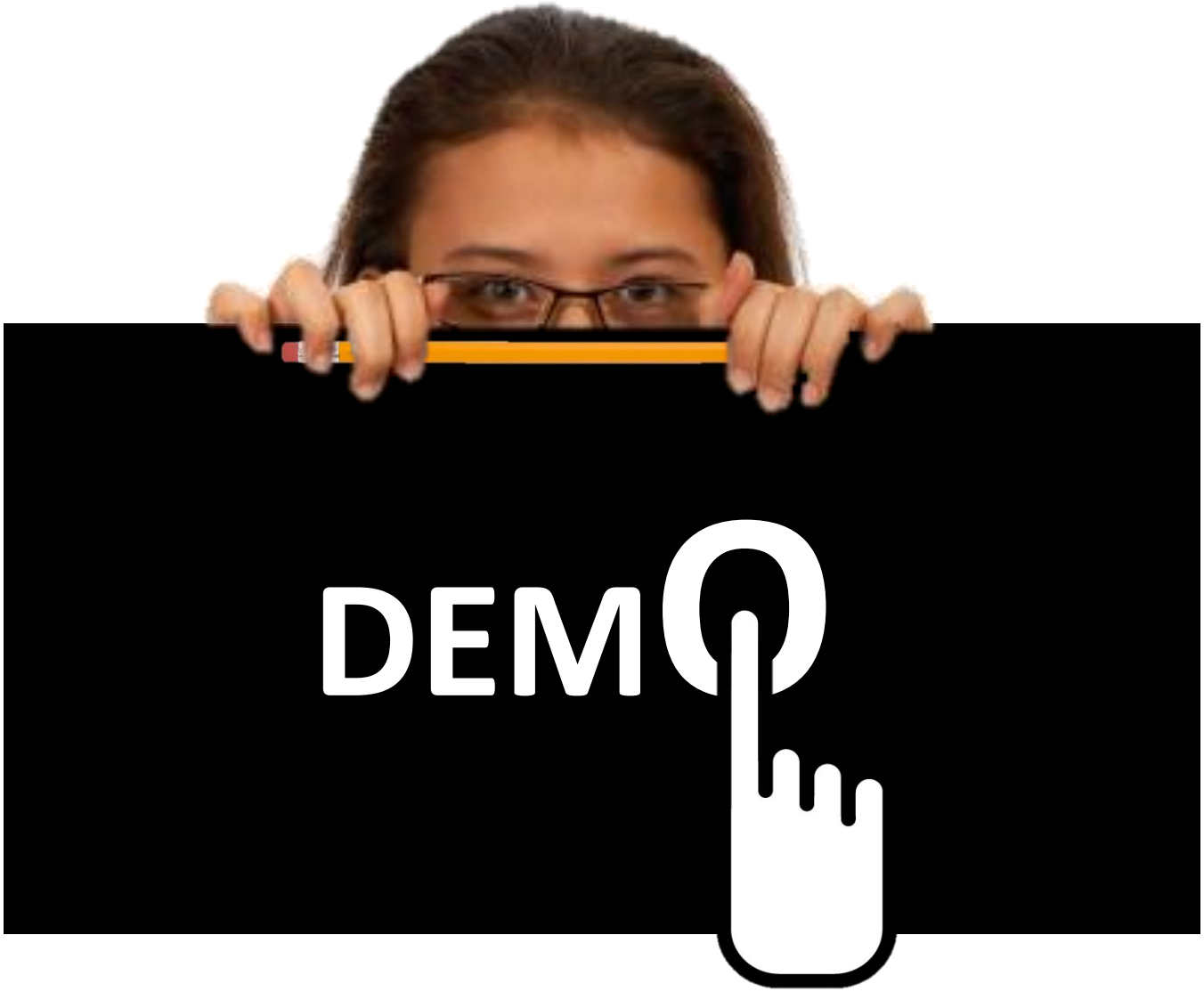


Demo 3:


DemoSOF203 - NetBeans IDE 8.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Projects
Services
Files
DemoSOF203
Source Packages
Slide1
Demo1AWT.java
Demo2Swing.java
JFrameDemo1.java
JPanel3.java
JPanelDemo1.java
Test Packages
Libraries
Test Libraries
JPanel3.java
Source
Design
History
jButton1
jButton2
jButton3
jButton4
jButton5
jButton7
jButton6
jButton8
jButton9
Palette
Swing Controls
Menu Bar
Label
Button
Toggle Button
Check Box
Radio Button
Button Group
Combo Box
List
Text Field
Text Area
Scroll Bar
Slider
Progress Bar
GridLayout - Navigator
Form JPanel3
Other Components
[JFrame]
GridLayout
jButton1 [JButton]
jButton2 [JButton]
jButton3 [JButton]
jButton4 [JButton]
jButton5 [JButton]
jButton7 [JButton]
jButton6 [JButton]
jButton8 [JButton]
jButton9 [JButton]
GridLayout - Properties
Properties
Columns 3
Horizontal Gap 0
Rows 3
Vertical Gap 0
GridLayout
Output
DemoSOF203 (run) #5
run:
HTTP Server Monitor
DemoSOF203 (run) #6
running...
(7 more...)
19:35
INS

- Demo 3:


jButton1
jButton2
jButton3
jButton4
jButton5
jButton7
jButton6
jButton8
jButton9


DEMO

TỔNG KẾT

- Giới thiệu gói thư viện AWT
- Giới thiệu SWING
- So sánh AWT và Swing
- Container trong Java Swing
- Container Component (JFrame, JPanel)
- Demo


TIN HỌC VĂN PHÒNG QUỐC TẾ THEO CHUẨN ICDL
Syllabus 5 - Office 2007
LUẬT KINH TẾ
LUẬT KINH TẾ
KỸ THUẬT SOẠN THẢO VÀ TRÌNH BÀY VĂN BẢN
TIN HỌC VĂN PHÒNG QUỐC TẾ THEO CHUẨN ICDL
Syllabus 5 - Office 2007
Cảm ơn