

Bài 02:

LỚP JFRAME, JPANEL, JLABEL, JTEXTFIELD VÀ JBUTTON

Giảng Viên: ThS. Giang Hào Côn

Lớp JFrame

- JFrame là một cửa sổ được sử dụng để tạo các ứng dụng GUI độc lập. JFrame có thanh tiêu đề, đường viền và các nút để đóng, thu nhỏ và phóng to cửa sổ.
- Một JFrame có thể đặt bên trong nó nhiều khung (Panel) và các thành phần con như: nút lệnh (Button), nhãn (Label), khung nhập (TextField),...
- Cú pháp tạo một JFrame trong Swing:

`JFrame <tên_biến> = new JFrame("...");`

Lớp JFrame

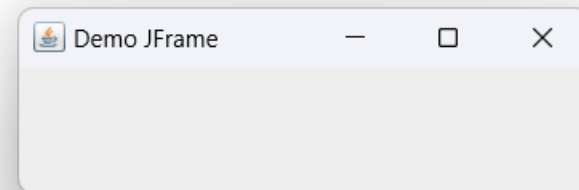
Mặc định khi một JFrame được khởi tạo xong nó sẽ ở trạng thái ẩn, tức là người dùng sẽ không thấy được. Nếu muốn hiển thị JFrame này đến người dùng chúng ta cần gọi đến **setVisible(boolean visible)** với giá trị truyền vào:

- **True** – JFrame sẽ hiện thị ra giao diện.
- **False** – JFrame sẽ bị ẩn đi.

```
import javax.swing.JFrame;

public class Vidu03 {

    public static void main(String[] args) {
        // Create a frame
        JFrame frame = new JFrame("Demo JFrame");
        frame.setSize(300, 100);
        // Display the frame
        frame.setVisible(true);
    }
}
```



Lớp JFrame

Xử lý khi JFrame bị đóng

```
public static void main(String[] args) {  
    // Create a frame  
    JFrame frame = new JFrame("Demo JFrame");  
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  
    frame.setSize(300, 100);  
    // Display the frame  
    frame.setVisible(true);  
}
```

Sau khi JFrame bị đóng, chương trình cũng sẽ tắt theo.

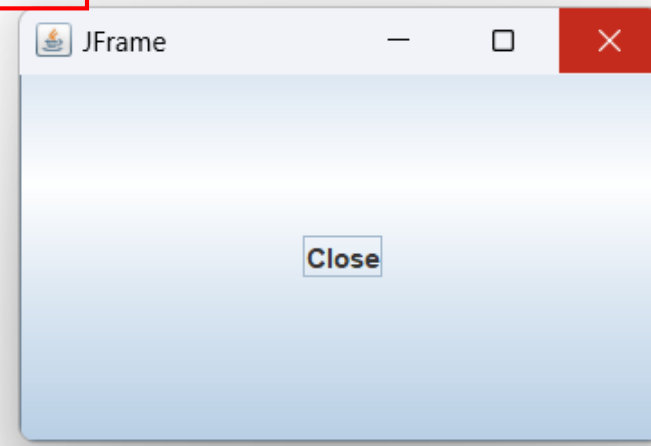
Lớp JFrame

Chiều cao, rộng và vị trí của JFrame trên màn hình Desktop

```
public static void main(String[] args) {  
    // Create a frame  
    JFrame frame = new JFrame("JFrame");  
    frame.setSize(300, 100);  
    frame.setLocation(500, 500);  
    // Display the frame  
    frame.setVisible(true);  
}
```

Thêm các component con vào JFrame

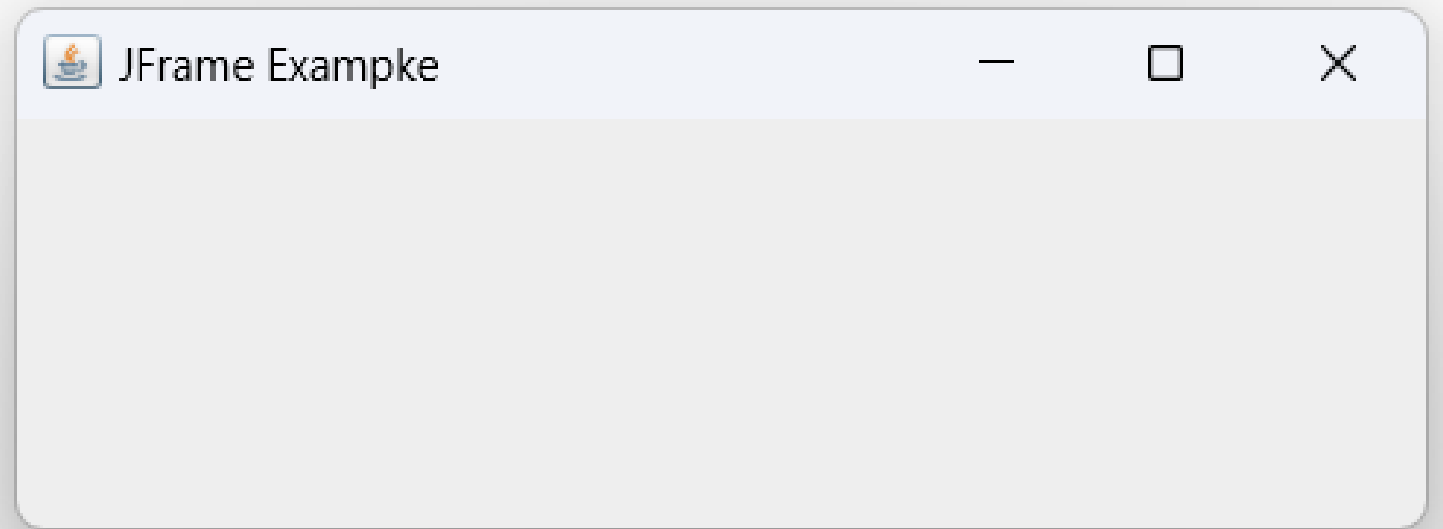
```
import java.awt.Container;  
import javax.swing.JButton;  
import javax.swing.JFrame;  
  
public class Vidu03 {  
  
    public static void main(String[] args) {  
        JFrame frame = new JFrame("JFrame");  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        Container contentPane = frame.getContentPane();  
        // Add a close button  
        JButton closeButton = new JButton("Close");  
        contentPane.add(closeButton);  
        // set the size of the frame 300 x 200  
        frame.setBounds(50, 50, 300, 200);  
        frame.setVisible(true);  
    }  
}
```



Lớp JFrame

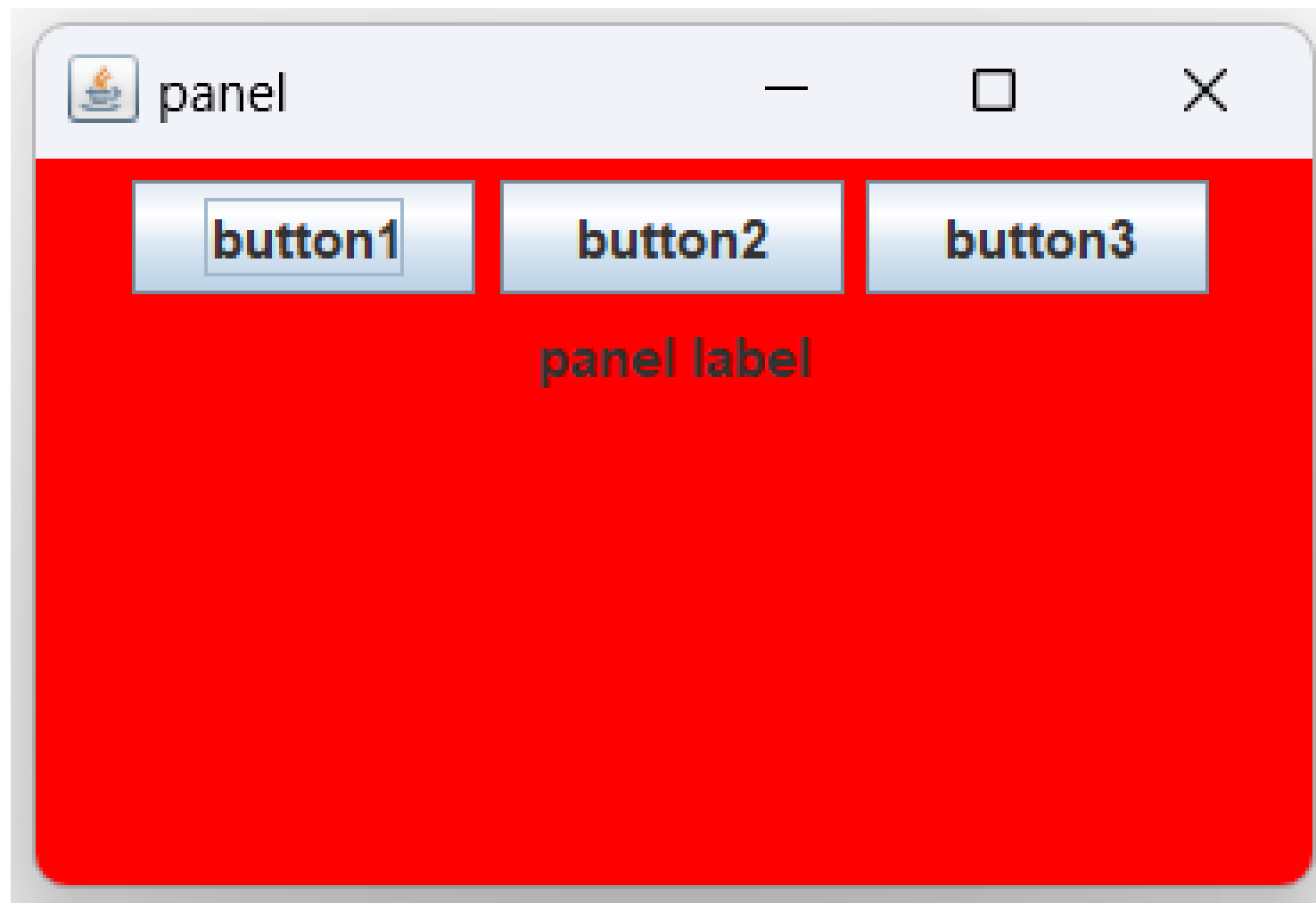
Canh JFrame ở giữa màn hình

```
public static void main(String[] args) {  
    JFrame mainFrame = new JFrame("JFrame Exampke");  
    mainFrame.setSize(400, 150);  
    mainFrame.setLocationRelativeTo(null);  
    mainFrame.setVisible(true);  
}
```



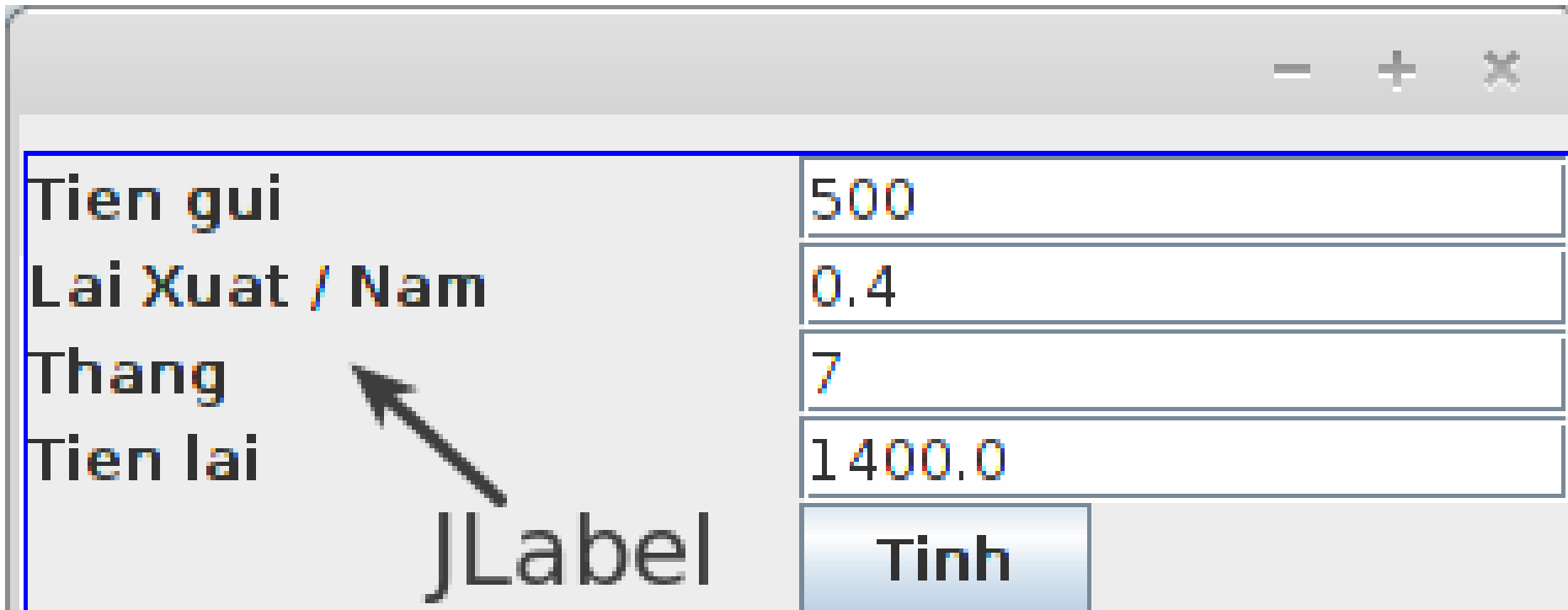
- **JPanel** là một container trong Swing dùng để chứa và sắp các các component khác bên trong nó. *Nhiệm vụ chính của JPanel là tổ chức các thành phần, nhiều bố cục khác nhau có thể được thiết lập trong JPanel giúp tổ chức các thành phần tốt hơn*, tuy nhiên nó không có thanh tiêu đề như JFrame và thường được dùng trong JFrame.
- **Sự khác biệt chính giữa JPanel và JFrame là:**
 - JFrame là một cửa sổ được sử dụng để tạo các ứng dụng GUI độc lập. Ngược lại, JPanel là nơi người ta có thể lắp ráp các thành phần hoặc thao tác phức tạp.
 - JFrame có thể chứa trong chính nó nhiều JPanel. Nhưng JPanel không thể chứa bên trong JFrames. Tuy nhiên, nó có thể đặt trong chính nó nhiều hoạt động.

Ví dụ



JLabel

- **JLabel** thường được dùng để hiển thị text hoặc hình ảnh để tạo các chỉ dẫn, hướng dẫn trên giao diện người dùng.



Tien gui	500
Lai Xuat / Nam	0.4
Thang	7
Tien lai	1400.0
<input type="button" value="Tinh"/>	

JLabel

JLabel

■ Tạo JLabel trong Java Swing

```
JLabel label = new JLabel("Hello, World!");
```

Tên biến do người dùng tự đặt tên

Dòng chữ hiện trên mặt nhãn

■ Đặt font chữ cho JLabel trong Java Swing

```
label.setFont(new Font("Arial", Font.BOLD, 16))
```

- Đặt màu nền cho JLabel trong Java Swing

```
label.setForeground(Color.RED)
```

- Đặt hình ảnh cho JLabel trong Java Swing

```
label.setIcon(new ImageIcon("image.png"))
```

■ Canh chỉnh nội dung cho JLabel trong Java Swing

- Sử dụng phương thức `setHorizontalAlignment(int alignment)` để căn chỉnh văn bản theo chiều ngang. Các giá trị alignment có thể là `SwingConstants.LEFT` , `SwingConstants.CENTER` , `SwingConstants.RIGHT` . Ví dụ: `label.setHorizontalAlignment(SwingConstants.CENTER)` .
- Sử dụng phương thức `setVerticalAlignment(int alignment)` để căn chỉnh văn bản theo chiều dọc. Các giá trị alignment có thể là `SwingConstants.TOP` , `SwingConstants.CENTER` , `SwingConstants.BOTTOM` . Ví dụ: `label.setVerticalAlignment(SwingConstants.CENTER)` .

■ Đặt Text cho JLabel trong Java Swing

- Sử dụng phương thức `setText(String text)` để đặt nội dung văn bản cho JLabel. Ví dụ:
`label.setText("Hello, World!")` .
- Sử dụng phương thức `setToolTipText(String text)` để đặt một thông báo gợi ý cho JLabel. Ví dụ: `label.setToolTipText("This is a JLabel")` .

■ Đặt kích thước và vị trí cho JLabel trong Java Swing

- Sử dụng phương thức `setSize(int width, int height)` để đặt kích thước của JLabel. Ví dụ: `label.setSize(200, 100)` .
- Sử dụng phương thức `setLocation(int x, int y)` để đặt vị trí của JLabel trong container cha. Ví dụ: `label.setLocation(50, 50)` .

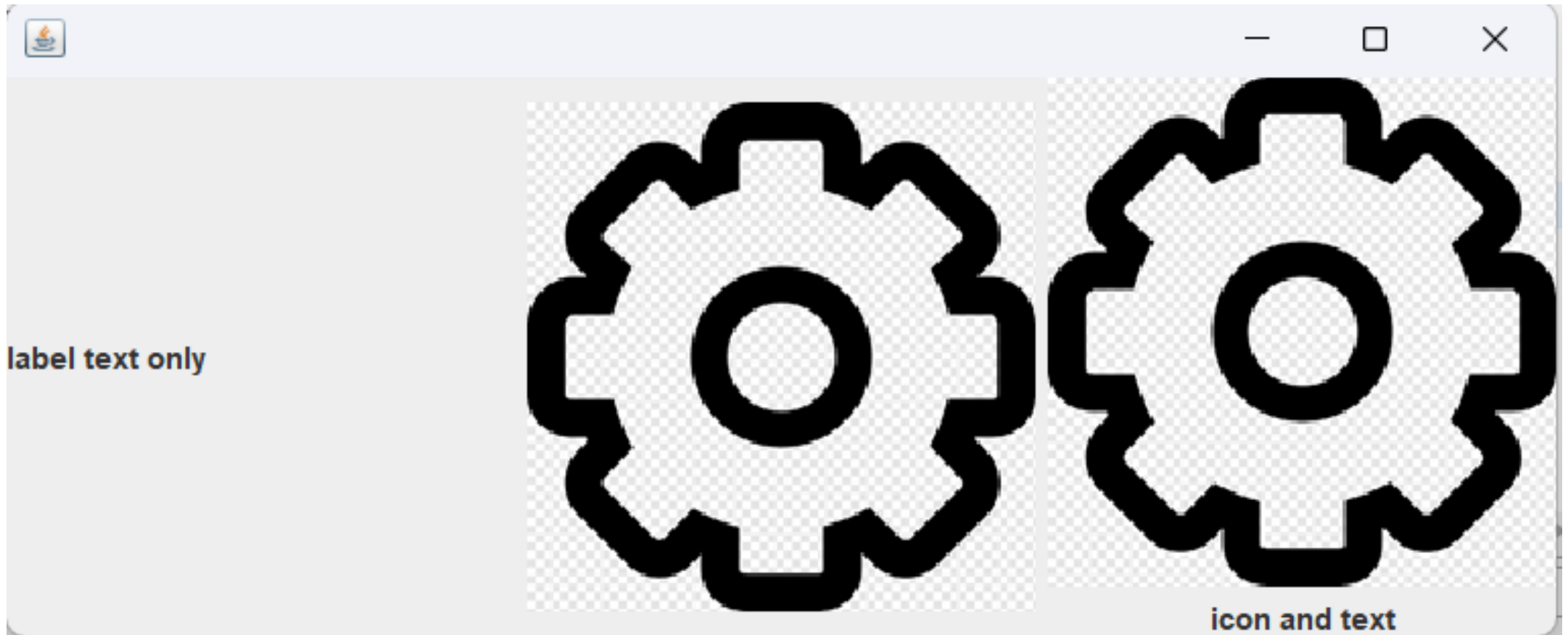
- Xử lý sự kiện cho JLabel trong Java Swing

```
// Xử lý sự kiện khi JLabel được nhấp chuột
label.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        JOptionPane.showMessageDialog(null, "Label clicked!");
    }
});
```

Hãy trình bày Ứng dụng của JLabel trong giao diện người dùng

JLabel

- Ví dụ: Tạo các JLabel đơn giản.



- Ví dụ: Đặt màu, màu nền cho JLabel .



TextField

- **TextField** là một đối tượng cho phép người dùng nhập một dòng văn bản. Thường dùng để nhập dữ liệu với các thông tin ngắn.

Tien gui	500
Lai Xuat / Nam	0.4
Thang	7
Tien lai	1400.0
TextField	
Tinh	

JTextField

- Tạo JTextField trong Java Swing.

```
JTextField textField = new JTextField();
```

Tên biến do người dùng tự đặt tên

- Đặt kích thước và vị trí JTextField trong Java Swing.

```
textField.setBounds(x, y, width, height);
```

- Đặt giá trị cho JTextField trong Java Swing.

```
textField.setText("Giá trị văn bản");
```

JTextField

- Lấy giá trị của JTextField trong Java Swing.

```
String text = textField.getText();
```

- Đặt màu nền cho JTextField trong Java Swing.

```
textField.setBackground(Color.WHITE);
```

- Đặt font chữ cho JTextField trong Java Swing.

```
textField.setFont(new Font("Arial", Font.PLAIN, 12));
```

```
textField.setFont(textField.getFont().deriveFont(Font.BOLD));
```

JTextField

- Vô hiệu hoá JTextField trong Java Swing.

```
textField.setEnabled(false);
```

- Đặt chiều dài tối đa cho JTextField trong Java Swing.

```
textField.setDocument(new JTextFieldLimit(10));
```

- Đặt hình dạng con trỏ cho JTextField trong Java Swing.

```
textField.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
```

JTextField

▪ Xử lý sự kiện cho JTextField trong Java Swing.

```
textField.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        // Xử lý sự kiện khi người dùng nhấn Enter trong JTextField  
    }  
});
```

▪ Xử lý sự kiện thay đổi nội dung JTextField trong Java Swing.

```
textField.getDocument().addDocumentListener(new DocumentListener() {  
    public void insertUpdate(DocumentEvent e) {  
        // Xử lý sự kiện khi có thay đổi nội dung  
        String text = textField.getText();  
        // ...  
    }  
  
    public void removeUpdate(DocumentEvent e) {  
        // Xử lý sự kiện khi có thay đổi nội dung  
        String text = textField.getText();  
        // ...  
    }  
  
    public void changedUpdate(DocumentEvent e) {  
        // Xử lý sự kiện khi có thay đổi nội dung  
        String text = textField.getText();  
        // ...  
    }  
});
```


JTextField

- Xử lý sự kiện khi JTextField nhận con trỏ và mất con trỏ.

```
textField.addFocusListener(new FocusListener() {  
    public void focusGained(FocusEvent e) {  
        // Xử lý sự kiện khi TextField được tập trung vào  
        // ...  
    }  
  
    public void focusLost(FocusEvent e) {  
        // Xử lý sự kiện khi TextField mất tập trung  
        // ...  
    }  
});
```

JTextField

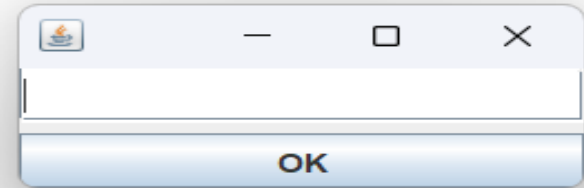
▪ Xử lý sự kiện khi phím được nhấn trên JTextField

```
textField.addKeyListener(new KeyListener() {  
    public void keyPressed(KeyEvent e) {  
        // Xử lý sự kiện khi có phím được nhấn  
        // ...  
    }  
  
    public void keyReleased(KeyEvent e) {  
        // Xử lý sự kiện khi phím được nhả ra  
        // ...  
    }  
  
    public void keyTyped(KeyEvent e) {  
        // Xử lý sự kiện khi có phím được gõ  
        // ...  
    }  
});
```

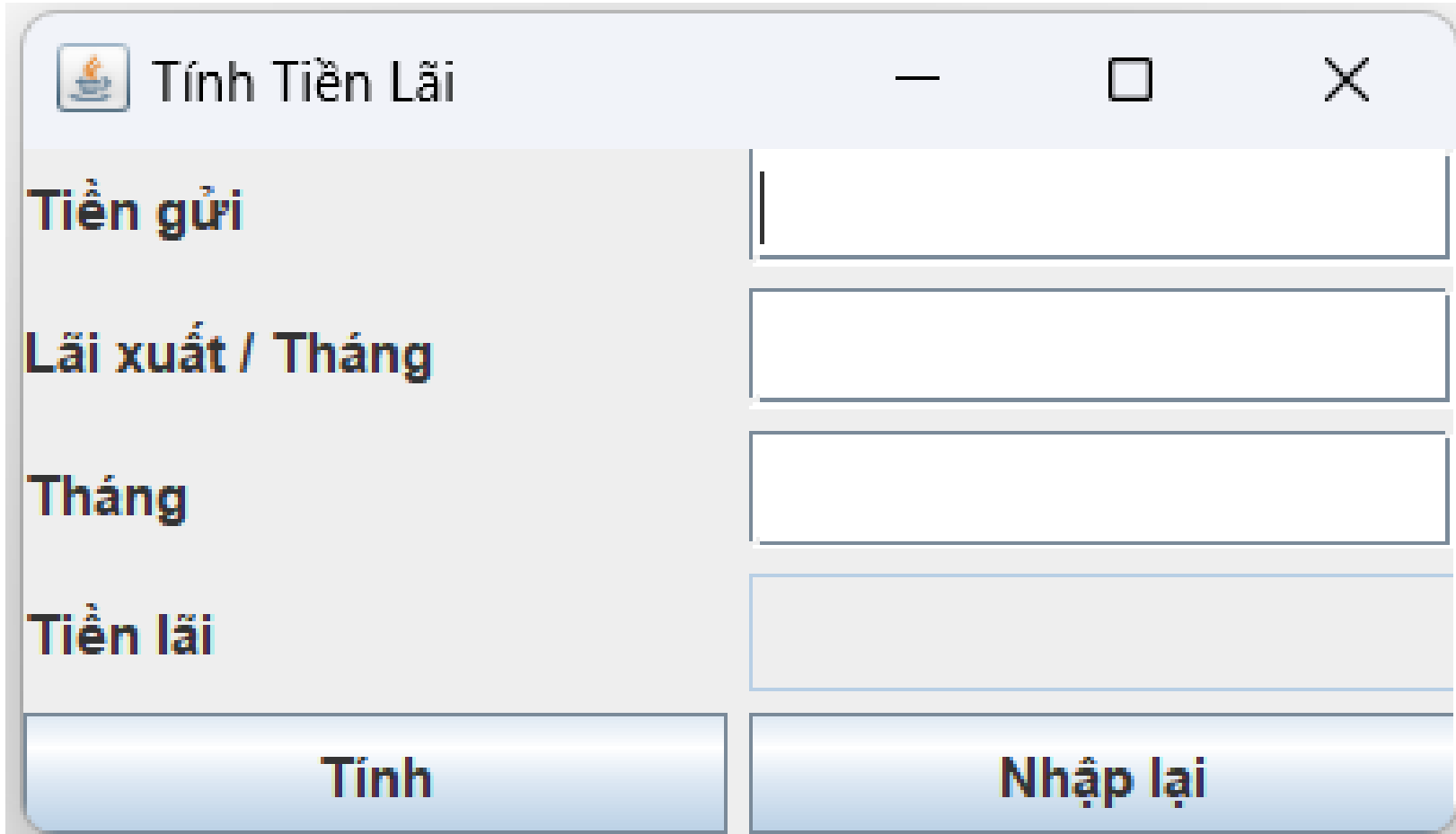
JTextField


▪ Ví dụ 01: Tạo JTextField đơn giản.

```
public class MyJTextField extends JFrame {  
    public MyJTextField() {  
        // create JFrame  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        setLayout(new GridLayout(2, 1, 5, 5));  
  
        // create and add JTextField  
        JTextField tf = new JTextField(20);  
        add(tf);  
  
        // add a JButton  
        add(new JButton("OK"));  
  
        // Display JFrame  
        pack();  
        setLocationRelativeTo(null);  
        setVisible(true);  
    }  
  
    public static void main(String[] args) {  
        new MyJTextField();  
    }  
}
```



▪ Ví dụ 02: Tính tiền lãi.



 **Tính Tiền Lãi** — □ ×

Tiền gửi

Lãi xuất / Tháng

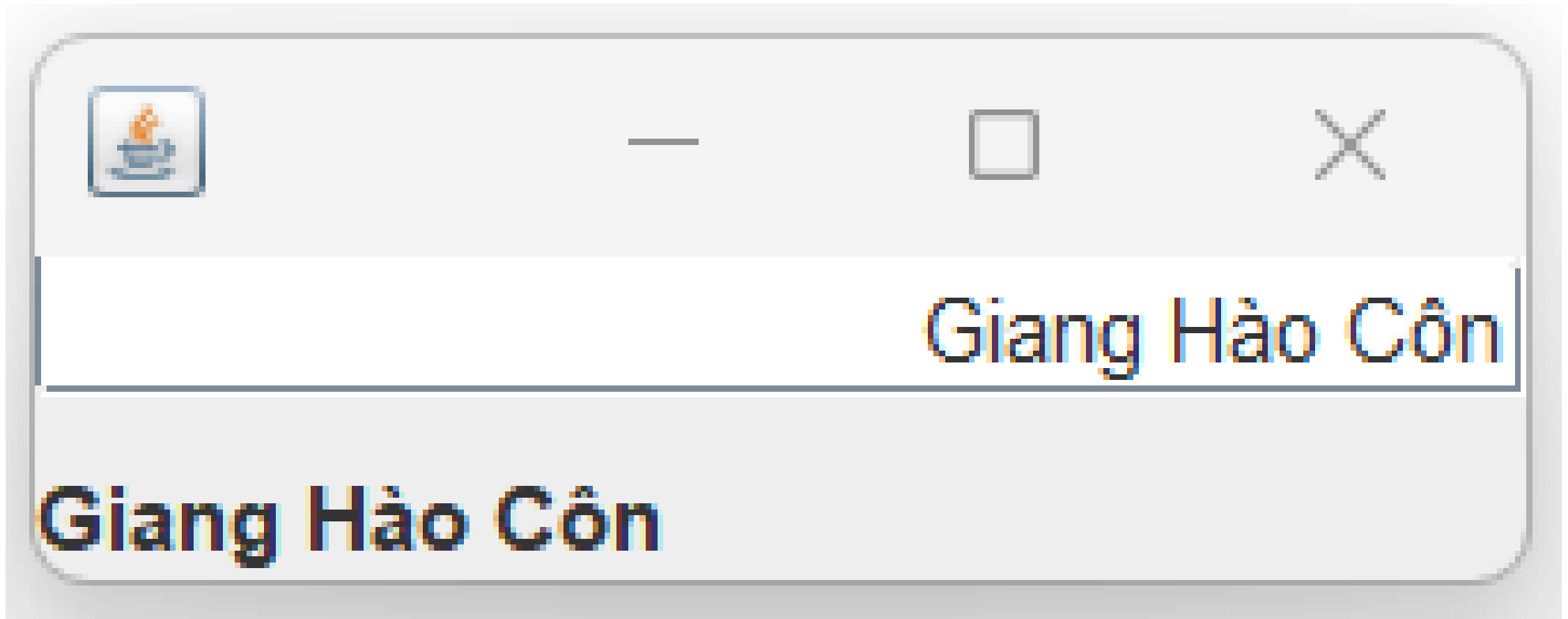
Tháng

Tiền lãi

Tính **Nhập lại**

JTextField

- Ví dụ 03: Căn lề và tạo sự kiện cho JTextField.



Hãy trình bày Ứng dụng của JTextField trong giao diện người dùng

JPasswordField



JPasswordField

- Tạo JPasswordField trong Java Swing.

```
JPasswordField passwordField = new JPasswordField();
```

Tên biến do người dùng tự đặt tên

- Đặt kích thước JPasswordField trong Java Swing.

```
passwordField.setBounds(x, y, width, height);
```


JPasswordField

- **Lấy giá trị của JPasswordField trong Java Swing.**

```
char[] password = passwordField.getPassword();  
String passwordString = new String(password);
```

- **Đặt màu chữ cho JPasswordField**

```
passwordField.setForeground(Color.RED);
```

JPasswordField

▪ Đặt màu nền cho JPasswordField

```
passwordField.setBackground(Color.YELLOW);
```

▪ Đặt font chữ cho JPasswordField

```
Font font = new Font("Arial", Font.BOLD, 16); // Tạo một font chữ  
passwordField.setFont(font); // Thiết lập font chữ và kích thước
```

JPasswordField

▪ Đặt viền cho JPasswordField

```
passwordField.setBorder(BorderFactory.createLineBorder(Color.BLUE));
```

▪ Đặt tooltip (chú thích) cho JPasswordField

```
passwordField.setToolTipText("Nhập mật khẩu");
```

JPasswordField

▪ Ví dụ xử lý sự kiện cho JPasswordField

```
passwordField.setEchoChar((char) 0); // Ẩn ký tự hiển thị của mật khẩu
passwordField.setText("Nhập mật khẩu"); // Đặt văn bản gợi ý
passwordField.addFocusListener(new FocusAdapter() {
    public void focusGained(FocusEvent e) {
        if (String.valueOf(passwordField.getPassword()).equals("Nhập mật khẩu")) {
            passwordField.setText("");
            passwordField.setEchoChar('*'); // Đặt ký tự hiển thị của mật khẩu
        }
    }
    public void focusLost(FocusEvent e) {
        if (String.valueOf(passwordField.getPassword()).isEmpty()) {
            passwordField.setText("Nhập mật khẩu");
            passwordField.setEchoChar((char) 0);
        }
    }
});
```

■ Ví dụ 01

```
import javax.swing.*;
public class PasswordFieldExample {
    public static void main(String[] args) {
        JFrame f=new JFrame("Password Field Example");
        JPasswordField value = new JPasswordField();
        JLabel l1=new JLabel("Password:");
        l1.setBounds(20,100, 80,30);
        value.setBounds(100,100,100,30);
        f.add(value);  f.add(l1);
        f.setSize(300,300);
        f.setLayout(null);
        f.setVisible(true);
    }
}
```

Password:

JPasswordField

▪ Ví dụ 02: Jpasswordfield và ActionListener

```
import javax.swing.*;
import java.awt.event.*;
public class PasswordFieldExample {
    public static void main(String[] args) {
        JFrame f=new JFrame("Password Field Example");
        final JLabel label = new JLabel();
        label.setBounds(20,150, 200,50);
        final JPasswordField value = new JPasswordField();
        value.setBounds(100,75,100,30);
        JLabel l1=new JLabel("Username:");
        l1.setBounds(20,20, 80,30);
        JLabel l2=new JLabel("Password:");
        l2.setBounds(20,75, 80,30);
        JButton b = new JButton("Login");
        b.setBounds(100,120, 80,30);
        final JTextField text = new JTextField();
        text.setBounds(100,20, 100,30);
        f.add(value); f.add(l1); f.add(label); f.add(l2); f.add(b); f.add(text);
        f.setSize(300,300);
        f.setLayout(null);
        f.setVisible(true);
        b.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String data = "Username " + text.getText();
                data += ", Password: ";
                + new String(value.getPassword());
                label.setText(data);
            }
        });
    }
}
```

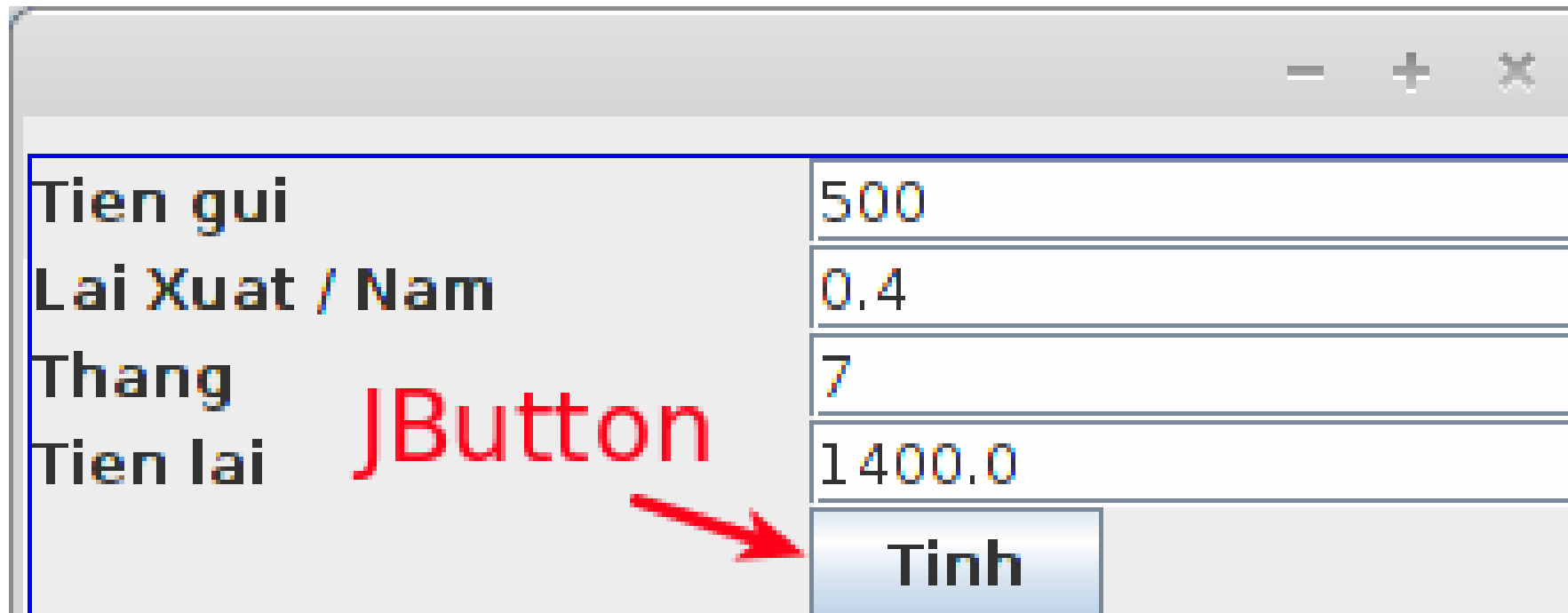
Username:

Password:

Username Nakul, Password: 1234

JButton

- **JButton** là một đối tượng mà cho phép chúng ta khi click chuột vào sẽ thực hiện một việc gì đó.



Tien gui	500
Lai Xuat / Nam	0.4
Thang	7
Tien lai	1400.0
Tinh	

JButton

■ tạo Button trong Java Swing.

```
JButton btn = new JButton("Click Button");
```

Tên biến do người dùng tự đặt tên

Dòng chữ hiện trên mặt nút

■ Tạo Button với Icon trong Java Swing

```
JButton button = new JButton(new ImageIcon("images/start.gif"));
```

Tên biến do người dùng tự đặt tên

Tập tin Hình hiện trên mặt nút

JButton

- tạo Button có hình và chữ trong Java Swing.

```
JButton button = new JButton("Start", new ImageIcon("images/start.gif"));
```

Tên biến do người dùng tự đặt tên

Dòng chữ hiện trên mặt nút

Tập tin Hình hiện trên mặt nút

JButton

- Tạo phím tắt, phím ghi nhớ cho Button



```
button.setMnemonic(KeyEvent.VK_E);
```

- Định màu nền, font chữ cho nút

```
button.setBackground(Color.YELLOW);
```

```
button.setFont(new java.awt.Font("Arial", Font.BOLD, 14));
```

Trong đó **button** là tên biến của nút lệnh

JButton

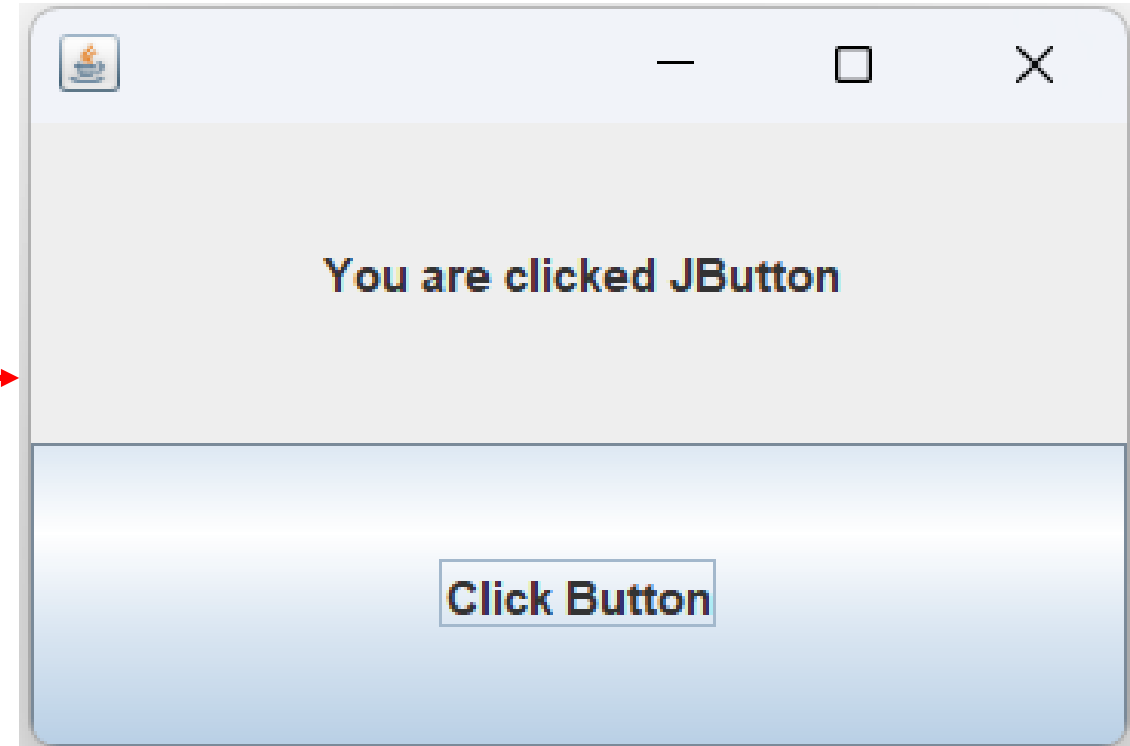
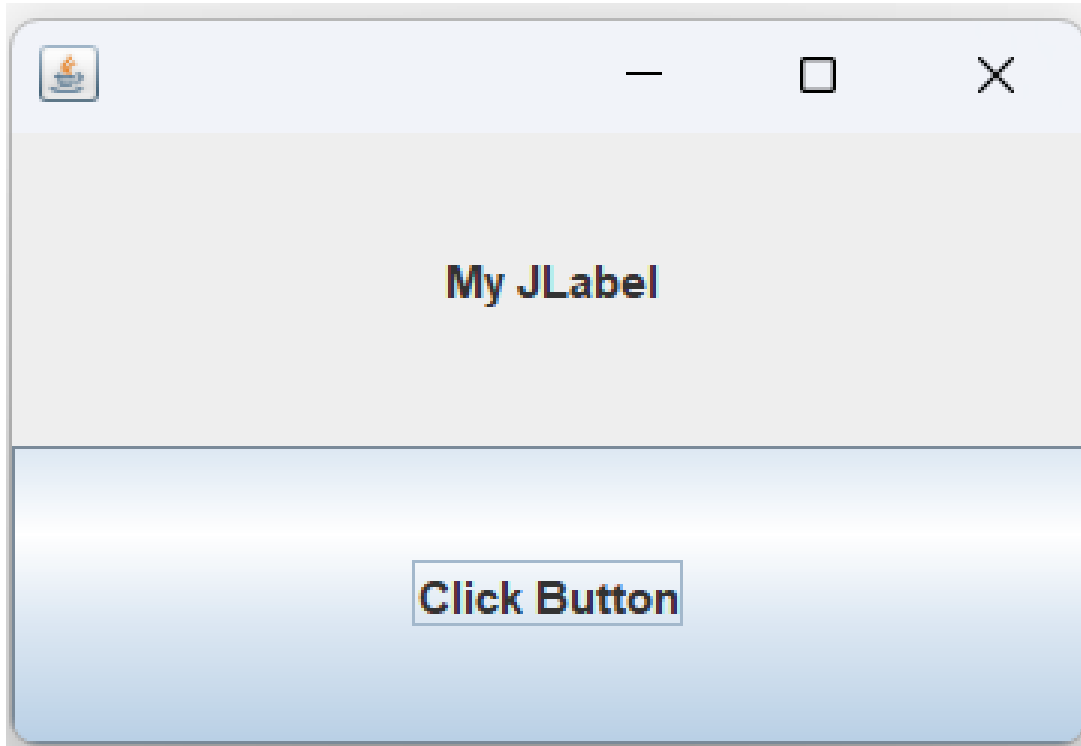
▪ Xử lý sự kiện click trong Button

```
button.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent evt) {  
        // do everything here...  
    }  
});
```

Trong đó **button** là tên biến của nút lệnh

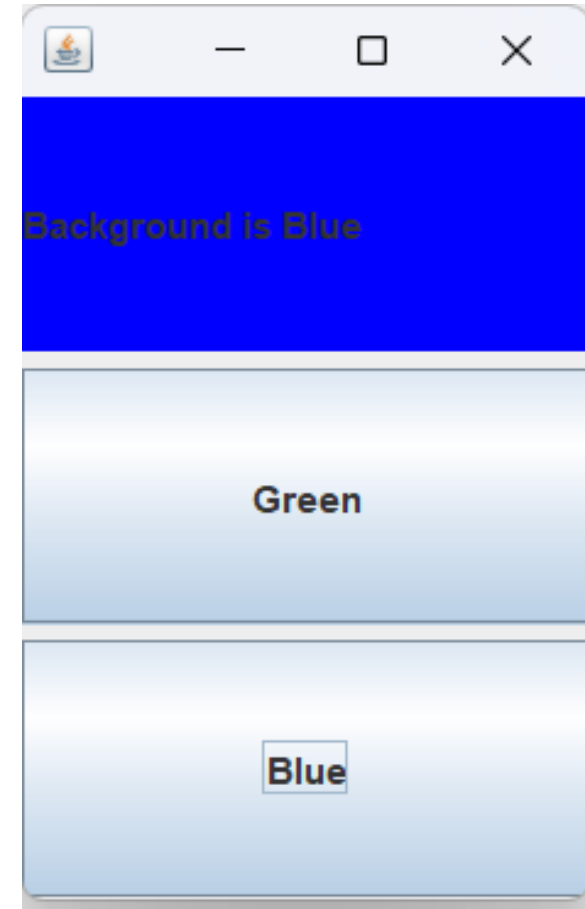
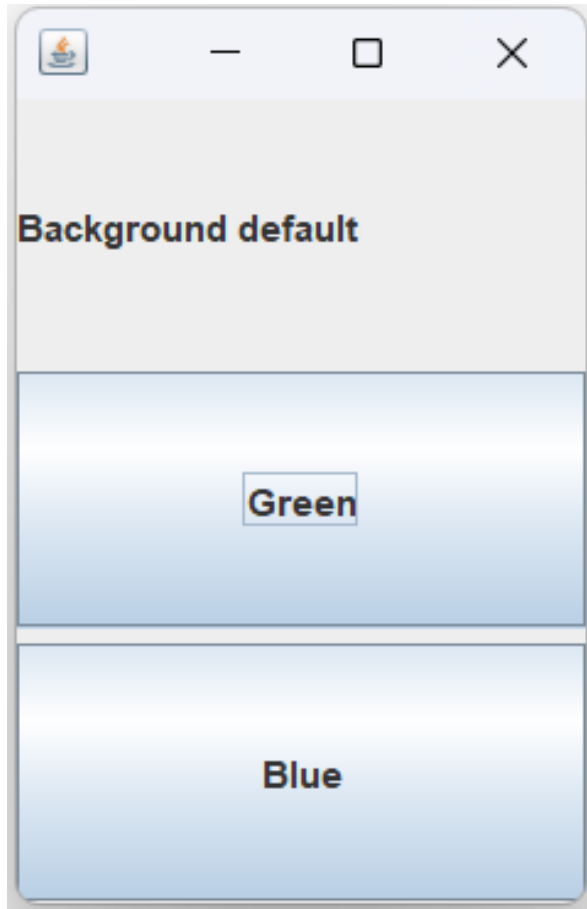
JButton

- Ví dụ 1: Tạo một JButton đơn giản.



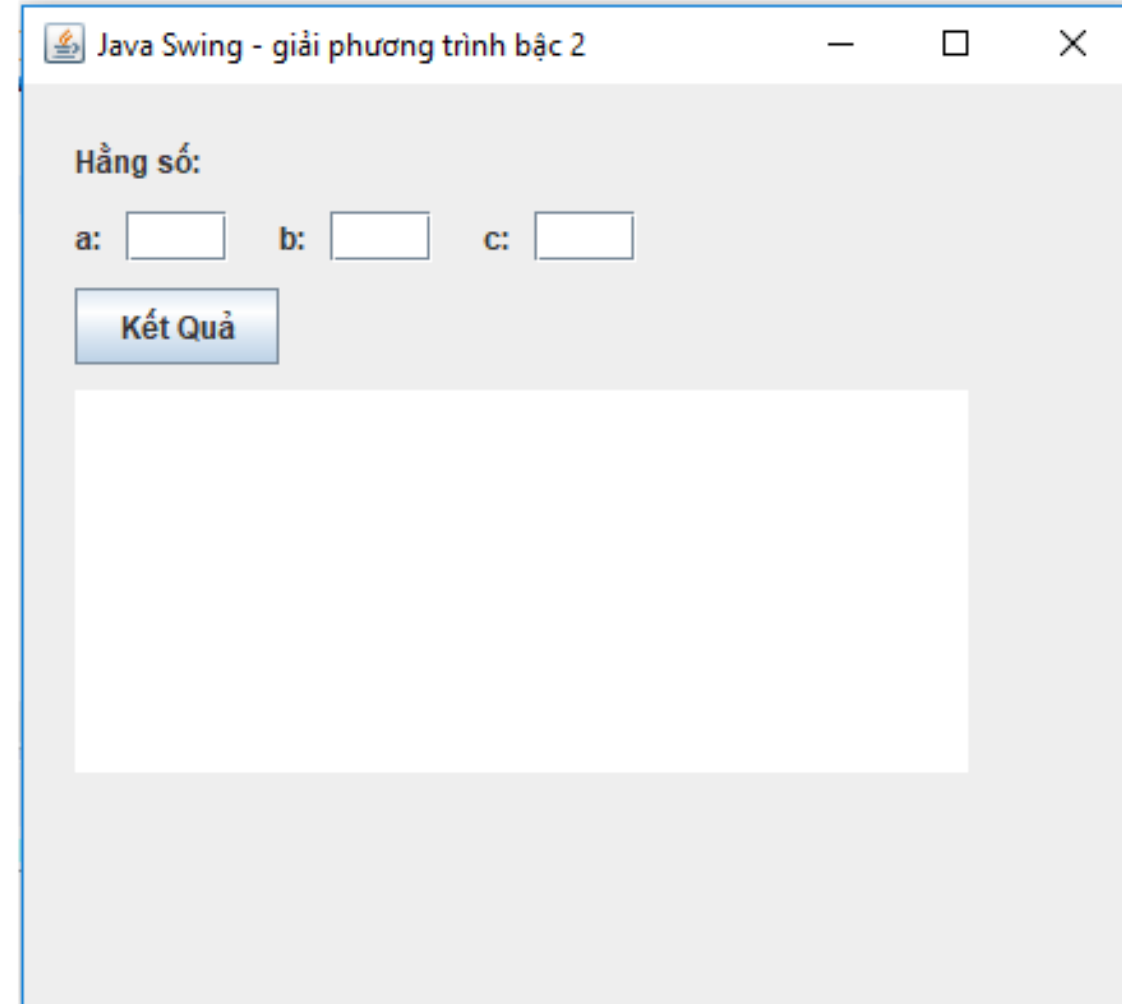
JButton

▪ Ví dụ 2: Cách tạo và bắt sự kiện JButton.



Câu hỏi thảo luận

- 1) Phân biệt JFrame và JPanel trong Java Swing.
- 2) Trình bày ứng dụng của JLabel, JTextField, JPasswordField, JButton trong UI.
- 3) Viết Code cho giao diện sau đây (sử dụng JFrame, JPanel, JLabel, JTextField và JButton):



Java Swing - giải phương trình bậc 2

Hằng số:

a: b: c: