

## MÓDULO 4 – SWING – Parte 3

### // Exemplo11 – Ação do Mouse

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class Ex11 extends JFrame implements ActionListener{

    private JPanel painel;
    private JButton botaoOK;
    private JButton botaoCancel;

    public Ex11()
    {
        super("Ação do Mouse");
        botaoOK = new JButton("OK");
        botaoOK.setToolTipText("Botão OK");
        //Mensagem exibida quando o mouse fica parado sobre o botão

        botaoCancel = new JButton("CANCELAR");
        botaoCancel.setToolTipText("Botão CANCELAR");

        //adiciona este botão, para sua ação ser tratada
        botaoOK.addActionListener(this);
        botaoCancel.addActionListener(this);

        painel = new JPanel(new GridLayout(2,1,30,10));
        painel.add(botaoOK);
        painel.add(botaoCancel);

        add(painel);
    }

    public void actionPerformed(ActionEvent e) { //método captura ação do mouse

        if (e.getSource() == botaoOK)
            JOptionPane.showMessageDialog(painel, "PRESSIONOU OK");
        else
            JOptionPane.showMessageDialog(null, "PRESSIONOU CANCELAR");
    }

    public static void main(String[] args) {
        Ex11 ex = new Ex11();
        ex.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        ex.setLocation(400,200);
        ex.setSize(200, 150);
        ex.setVisible(true);
    }
}
```



### // Exemplo12 - Outro exemplo com aplicação do mouse

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.*;

public class Ex12 extends JFrame implements ActionListener{

    private JButton botao1, botao2;
    private JPanel panel;

    public Ex12 ()
    {
        super("Ação do Mouse");

        botao1 = new JButton("PLAY");
        botao2 = new JButton("LIMPAR");
        panel = new JPanel();           // criação do Painei Principal

        panel.add(botao1);             // adiciona botão no panel
        botao1.addActionListener(this);
        botao2.addActionListener(this);

        getContentPane().add(panel);
        setSize(200,100);
        setVisible(true);
    }

    public static void main(String[] args)
    {
        Ex12 ex = new Ex12();
        ex.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

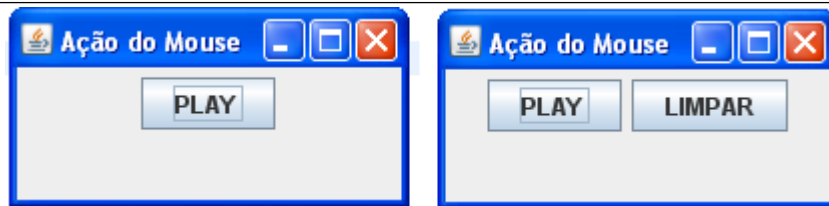
    public void actionPerformed(ActionEvent e) {
        if (e.getSource()==botao1){
            panel.add(botao2);
        }
        else
        {
            panel.remove(botao2);
        }
    }
}
```

```

    }

    this.repaint();
    setVisible(true);
}
}

```



### // Exemplo13 - Password.java

```

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;

public class Ex13 extends JFrame {

    private JTextField textField1;
    private JTextField textField2;
    private JTextField textField3;
    private JPasswordField passwordField;
    private JLabel label;

    public Ex13()
    {
        super( "Testando JTextField e JPasswordField" );
        setLayout( new FlowLayout() ); // define layout do frame

        // Construindo textfield com 10 colunas e sem texto
        textField1 = new JTextField( 10 );
        add( textField1 ); // add textField1 to JFrame

        // Construindo textfield com texto
        textField2 = new JTextField( "Digite um texto aqui" );
        add( textField2 ); // add textField2 to JFrame

        // Construindo textfield com texto default e 21 colunas
        textField3 = new JTextField( "TextField desabilitado para edição",21 );
        textField3.setEditable( false ); // desabilita edição
        add( textField3 ); // adiciona textField3

        // Construindo passwordfield com texto default
        passwordField = new JPasswordField( "Password" );
        add( passwordField ); // add passwordField to JFrame

        label = new JLabel("Tecle Enter após digitação em algum TextFields");
        add (label);

        // Registra evento handlers
        TextFieldHandler handler = new TextFieldHandler();
        textField1.addActionListener( handler );
        textField2.addActionListener( handler );
        textField3.addActionListener( handler );
        passwordField.addActionListener( handler );
    } //fim do construtor
}

```

```

private class TextFieldHandler implements ActionListener
{
    public void actionPerformed((ActionEvent event) )
    {
        String string = "";        // declaração de uma string para ser exibida

        // pressionando Enter no campo textField1
        if ( event.getSource() == textField1 )
            string = String.format("textField1: %s", event.getActionCommand());

        // pressionando Enter no campo textField2
        else if ( event.getSource() == textField2 )
            string = String.format( "textField2: %s", event.getActionCommand());

        // pressionando Enter no campo textField3
        else if ( event.getSource() == textField3 )
            string = String.format( "textField3: %s", event.getActionCommand());

        // pressionando Enter no campo passwordField
        else if ( event.getSource() == passwordField )
            string = String.format( "passwordField: %s", new String(
passwordField.getPassword() ) );

        // Exibindo conteúdo JTextField
        JOptionPane.showMessageDialog( null, string );

    } // fim do metodo actionPerformed

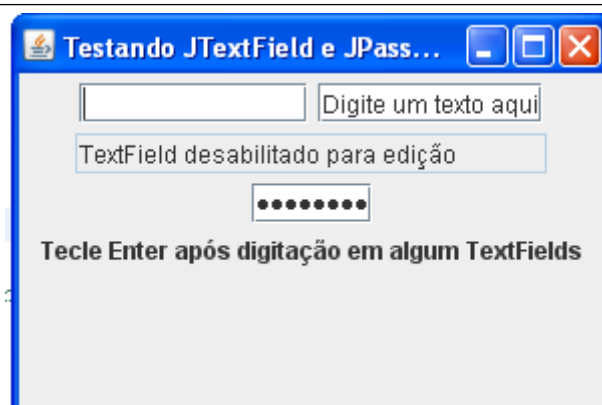
} // fim da classe TextFieldHandler

public static void main(String[] args){
    Ex13 ex = new Ex13();
    ex.setLocation(400,200);        //define a posição da janela no monitor
    ex.setSize(300,200);           //define o tamanho da janela
    ex.setVisible(true);
    ex.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

} //fim do void main

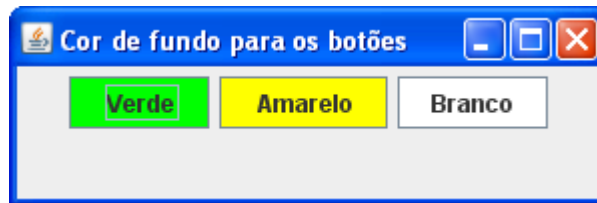
} // fim da classe Ex14

```

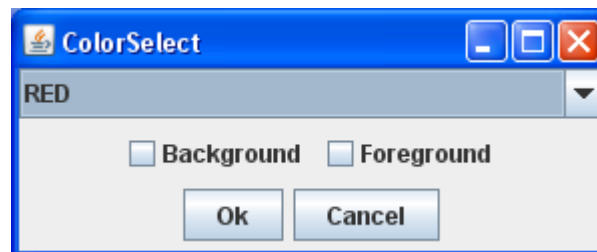


## Exercícios para Praticar

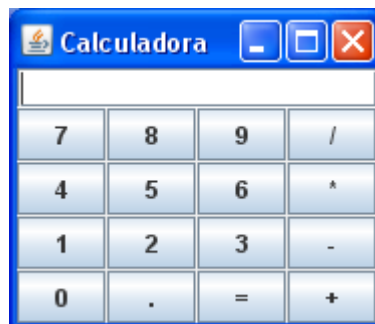
**01)** Implemente um programa em Java que resulte na seguinte janela e suas funcionalidades alterando a cor do fundo conforme a cor do botão clicado:



**02)** Implemente um programa em Java que resulte na seguinte janela e suas funcionalidades:



**03)** Implemente um programa em Java que resulte na seguinte janela e suas funcionalidades:



**04)** Implemente um programa em Java que resulte na seguinte janela e suas funcionalidades, sendo que caso o nome do usuário e senha forem válidos, exibir uma outra janela, caso contrário emitir uma mensagem informando dados incorretos:

