

Soft Blue Soft Blue Soft Blue Soft Blue Soft Blue

Soft Blue Soft Blue Soft Blue Soft Blue Soft Blue

Soft Blue Soft Blue Soft Blue Soft Blue Soft Blue

Java Avançado

Desenvolvendo Aplicações Gráficas com Swing

Soft Blue
www.softblue.com.br

Todos os direitos de cópia reservados. Não é permitida a distribuição física ou eletrônica deste material sem a permissão expressa e por escrito do autor.

Soft Blue Soft Blue Soft Blue Soft Blue Soft Blue

Soft Blue Soft Blue Soft Blue Soft Blue Soft Blue

Soft Blue Soft Blue Soft Blue Soft Blue Soft Blue

Tópicos Abordados

- Java Foundation Classes
 - Swing API
- Containers
- Components
- LayoutManagers
- Trabalhando com menus
- Tratamento de eventos
- Look & Feel

Soft Blue Soft Blue Soft Blue Soft Blue Soft Blue

Soft Blue Soft Blue Soft Blue Soft Blue Soft Blue

Soft Blue Soft Blue Soft Blue Soft Blue Soft Blue

Soft Blue Soft Blue Soft Blue Soft Blue Soft Blue

Soft Blue Soft Blue Soft Blue Soft Blue Soft Blue

Java Foundation Classes

- Conhecida como JFC
- É um conjunto de funcionalidades que proporciona a criação de interfaces gráficas (GUIs) com elementos gráficos e interatividade
- Os componentes **Swing** são um subconjunto do que a JFC disponibiliza para o programador

Soft Blue Soft Blue Soft Blue Soft Blue Soft Blue

Soft Blue Soft Blue Soft Blue Soft Blue Soft Blue

Swing

- O Swing possui uma série de componentes gráficos que permitem a criação de aplicações com interfaces gráficas ricas
 - Janelas
 - Painéis
 - Caixas de texto
 - Caixas de seleção
 - etc.
- Existe também suporte a eventos

Elementos do Swing

- Containers
 - Agrupam componentes
- Components
 - Componentes gráficos, como caixas de texto, ícones, caixas de seleção, etc.
- Layout Managers
 - Gerenciam a forma como os componentes devem ser organizados

Containers

- Os containers podem conter diversos componentes
- O mais usado é o **JFrame**

```
JFrame frame = new JFrame();
frame.setSize(300, 300);
frame.setLocation(100, 100);
frame.setTitle("Janela Principal");

frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setVisible(true);
```

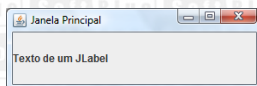
Components

- Componentes gráficos usados para criar a interface gráfica
- São inseridos dentro dos containers ou até dentro de outros componentes
- O Swing possui diversos tipos de componentes
 - Caixas de texto
 - Combo boxes
 - Radio buttons
 - Botões
 - etc.

JLabel

- Permite escrever textos

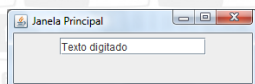
```
JLabel l = new JLabel();  
l.setText("Texto de um JLabel");  
frame.add(l);
```



JTextField

- Caixa de texto para entrada de dados

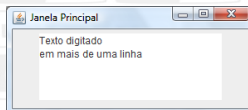
```
JTextField t = new JTextField();  
t.setText("Texto digitado");  
t.setColumns(15);  
frame.add(t);
```



JTextArea

- Caixa de texto para entrada de dados com suporte a múltiplas linhas

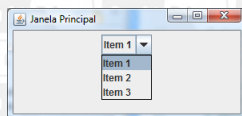
```
JTextArea t = new JTextArea();  
t.setText("Texto digitado\nem mais de uma linha");  
t.setRows(5);  
t.setColumns(20);  
frame.add(t);
```



JComboBox

- Caixa de seleção de itens

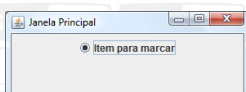
```
JComboBox c = new JComboBox();  
c.addItem("Item 1");  
c.addItem("Item 2");  
c.addItem("Item 3");  
frame.add(c);
```



JRadioButton

- Possibilita marcar determinado item de um conjunto de itens

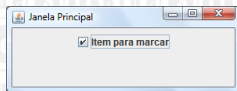
```
JRadioButton r = new JRadioButton();  
r.setText("Item para marcar");  
frame.add(r);
```



JCheckBox

- Possibilita marcar ou não determinado item

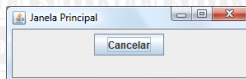
```
JCheckBox c = new JCheckBox();  
c.setText("Item para marcar");  
frame.add(c);
```



JButton

- Botão para ser pressionado

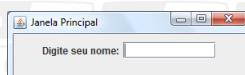
```
JButton b = new JButton();  
b.setText("Cancelar");  
frame.add(b);
```



JPanel

- Permite agrupar componentes

```
JPanel p = new JPanel();  
p.add(new JLabel("Digite seu nome:"));  
p.add(new JTextField(10));  
frame.add(p);
```

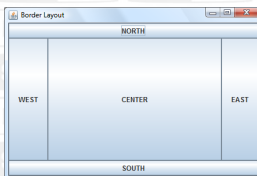


Layout Managers

- Permitem organizar os componentes dentro de containers ou outros componentes
- Tipos de layout managers
 - Border Layout
 - Flow Layout
 - Grid Layout
 - etc.

Border Layout

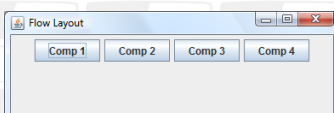
- Organiza os componentes em 5 regiões



```
f.setLayout(new BorderLayout());  
f.add(BorderLayout.NORTH, new JButton("NORTH"));  
f.add(BorderLayout.CENTER, new JButton("CENTER"));  
f.add(BorderLayout.SOUTH, new JButton("SOUTH"));  
f.add(BorderLayout.EAST, new JButton("EAST"));  
f.add(BorderLayout.WEST, new JButton("WEST"));
```

Flow Layout

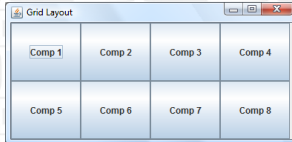
- Organiza os componentes em sequência



```
f.setLayout(new FlowLayout());  
f.add(new JButton("Comp 1"));  
f.add(new JButton("Comp 2"));  
f.add(new JButton("Comp 3"));  
f.add(new JButton("Comp 4"));
```

GridLayout

- Dispõe os componentes em grade



```
f.setLayout(new GridLayout(2, 4));
f.add(new JButton("Comp 1"));
f.add(new JButton("Comp 2"));
f.add(new JButton("Comp 3"));
f.add(new JButton("Comp 4"));
f.add(new JButton("Comp 5"));
f.add(new JButton("Comp 6"));
f.add(new JButton("Comp 7"));
f.add(new JButton("Comp 8"));
```

JMenuBar, JMenu e JMenuItem

- Permite criar uma barra de menu no container

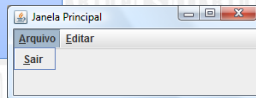
```
JMenuBar mb = new JMenuBar();

JMenu m1 = new JMenu("Arquivo");
m1.setMnemonic('A');

JMenuItem mil = new JMenuItem("Sair");
mil.setMnemonic('S');
m1.add(mil);

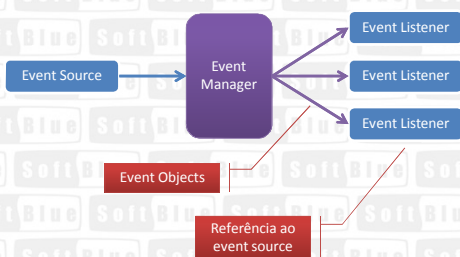
JMenu m2 = new JMenu("Editar");
m2.setMnemonic('E');

mb.add(m1);
mb.add(m2);
frame.setJMenuBar(mb);
```

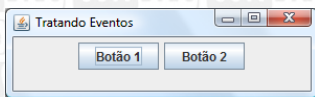


Arquitetura de Eventos

- O modelo de eventos em Swing é composto por **event sources** e **event listeners**



Exemplo de Tratamento de Evento

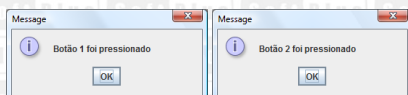


```
...  
AbrirCaixa action = new AbrirCaixa();  
JButton b1 = new JButton("Botão 1");  
b1.addActionListener(action);  
JButton b2 = new JButton("Botão 2");  
b2.addActionListener(action);  
...
```

Exemplo de Tratamento de Evento

```
public class AbrirCaixa implements ActionListener {  
    public void actionPerformed(ActionEvent e) {  
        JButton b = (JButton) e.getSource();  
        String nome = b.getText();  
        JOptionPane.showMessageDialog(null,  
            nome + " foi pressionado");  
    }  
}
```

Event Object



Exemplo de Tratamento de Evento

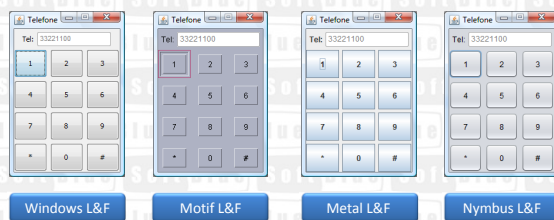
- É bastante comum o uso de classes anônimas no tratamento de eventos em Swing

```
...  
JButton b1 = new JButton("Botão 1");  
b1.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        //código de tratamento do evento  
    }  
});  
...
```


Look And Feel (L&F)

- A arquitetura do Swing permite o uso do que chamamos de look and feel
- O look and feel muda a forma como os componentes são visualizados (look) e como eles se comportam (feel)
- O look and feel pode ser obtido em tempo de execução dependendo da plataforma onde a aplicação está executando
- Pode ser mudado via programação

Exemplos de Look And Feel



Alterando o Look And Feel

- O look and feel pode ser alterado via programação de forma bastante simples

```
UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
```

Colocando em Prática...



Agora que você já aprendeu a teoria, acesse as vídeo-aulas práticas e pratique os assuntos abordados neste módulo!

[Clique aqui para acessar as vídeo-aulas práticas](#)
