

Roteiro - Painéis Java

Na linguagem java, dentro da categoria de interfaces gráficas, temos presentes alguns componentes que auxiliam na organização, design e funcionamento dos frames. São eles os painéis.

Utilizados a partir dos três pacotes gráficos da linguagem, servem como facilitador na criação de interfaces interativas dinâmicas e intuitivas para usuários.

Os pacotes em questão são:

- Abstract Window Toolkit (AWT);
- Standard Widget Toolkit (SWT);
- Swing.

Os componentes que escolhemos foram:

- JPanel;
- JTabbedPane.

Ambos pertencem ao pacote Swing.

Como são utilizados os componentes

Os componentes que escolhemos são utilizados juntos aos da classe JFrame, utilizando os métodos do pacote java.swing para configurar suas dimensões de layout, cores, visibilidade ao usuário e quais outros componentes serão adicionados.

São seus principais métodos:

```
public void add(Component c) - Adiciona um componente ao painel.  
public void setSize(int altura,int largura) - Define as dimensões do painel.  
public void setLayout(LayoutManager m) - Define o gerenciador de layout para o componente.  
public void setVisible(boolean b) - Define a visibilidade do componente ao usuário.
```

JPanel

Utilizado para criar espaços onde serão recebidos outros componentes da interface, como botões ou receptores de texto, é um dos mais utilizados na criação dos painéis dos frames por este motivo.

É declarado da seguinte forma:

```
public class JPanel extends JComponent implements Accessible
```

Estas são as principais formas que seu construtor é feito:

JPanel() - Utilizado para criar um novo JPanel com um buffer duplo e um layout de fluxo.

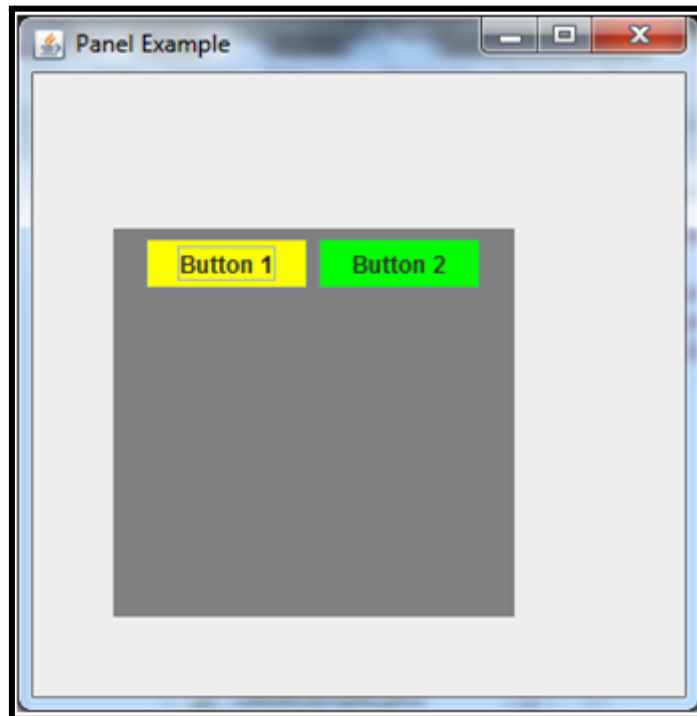
JPanel(boolean isDoubleBuffered) - Utilizado para criar um novo JPanel com FlowLayout e a estratégia de buffer especificada.

JPanel(LayoutManager layout) - Utilizado para criar um novo JPanel com o gerenciador de layout especificado.

E aqui um exemplo de código:

```
1. import java.awt.*;
2. import javax.swing.*;
3. public class PanelExample{
4.     PanelExample(){
5.         JFrame f= new JFrame("Panel Example");
6.         JPanel panel=new JPanel();
7.         panel.setBounds(40,80,200,200);
8.         panel.setBackground(Color.gray);
9.         JButton b1=new JButton("Button 1");
10.        b1.setBounds(50,100,80,30);
11.        b1.setBackground(Color.yellow);
12.        JButton b2=new JButton("Button 2");
13.        b2.setBounds(100,100,80,30);
14.        b2.setBackground(Color.green);
15.        panel.add(b1); panel.add(b2);
16.        f.add(panel);
17.        f.setSize(400,400);
18.        f.setLayout(null);
19.        f.setVisible(true);
20.    }
21.    public static void main(String args[])
22.    {
23.        new PanelExample();
24.    }
```

Output do código acima:



JTabbedPane

Utilizado para separar os demais componentes do painel em abas interativas, podendo nomeá-las títulos específicos.

É declarada da seguinte maneira:

```
public class JTabbedPane extends JComponent implements Serializable, Accessible, SwingConstants
```

Seu construtor é declarado das seguintes maneiras:

JTabbedPane() - Cria um JTabbedPane vazio com um posicionamento de guia padrão de JTabbedPane.Top.

JTabbedPane(int tabPlacement) - Cria um JTabbedPane vazio com o posicionamento de guia especificado.

JTabbedPane(int tabPlacement, int tabLayoutPolicy) - Cria um JTabbedPane vazio com um posicionamento de guia especificado e uma política de layout de guia.

E aqui um exemplo de código:

```
1. import javax.swing.*;
2. public class TabbedPaneExample {
3.     JFrame f;
4.     TabbedPaneExample(){
5.         f=new JFrame();
6.         JTextArea ta=new JTextArea(200,200);
7.         JPanel p1=new JPanel();
8.         p1.add(ta);
9.         JPanel p2=new JPanel();
10.        JPanel p3=new JPanel();
11.        JTabbedPane tp=new JTabbedPane();
12.        tp.setBounds(50,50,200,200);
13.        tp.add("main",p1);
14.        tp.add("visit",p2);
15.        tp.add("help",p3);
16.        f.add(tp);
17.        f.setSize(400,400);
18.        f.setLayout(null);
19.        f.setVisible(true);
20.    }
21.    public static void main(String[] args) {
22.        new TabbedPaneExample();
23.    }
```

Output do código:

