

INTERFACE JAVA PAINÉIS

- Jéssica Thais Xavier Alves
- João Victor dos Santos Nogueira



- Os painéis, presentes nas chamadas Interfaces Gráficas com Usuários (Graphical User Interface – GUI), **são componentes visuais que vão permitir ao usuário interagir com o programa** utilizando dispositivos externos, como mouse e teclado.

Atualmente, existem três bibliotecas gráficas presentes na linguagem Java. São elas:

- Abstract Window Toolkit (AWT)
 - Standard Widget Toolkit (SWT)
 - Swing
-
- Os pacotes adicionam componentes que permitem criar interfaces interativas através das chamadas de métodos e funções.



Alguns dos componentes presentes nos pacotes são:

- JLabel
- JTextField
- JComboBox
- JList
- JButton
- JCheckBox
- JFrame
- JPanel

Em prática, esses elementos são obtidos através da classe **Component** do pacote **java.awt** .

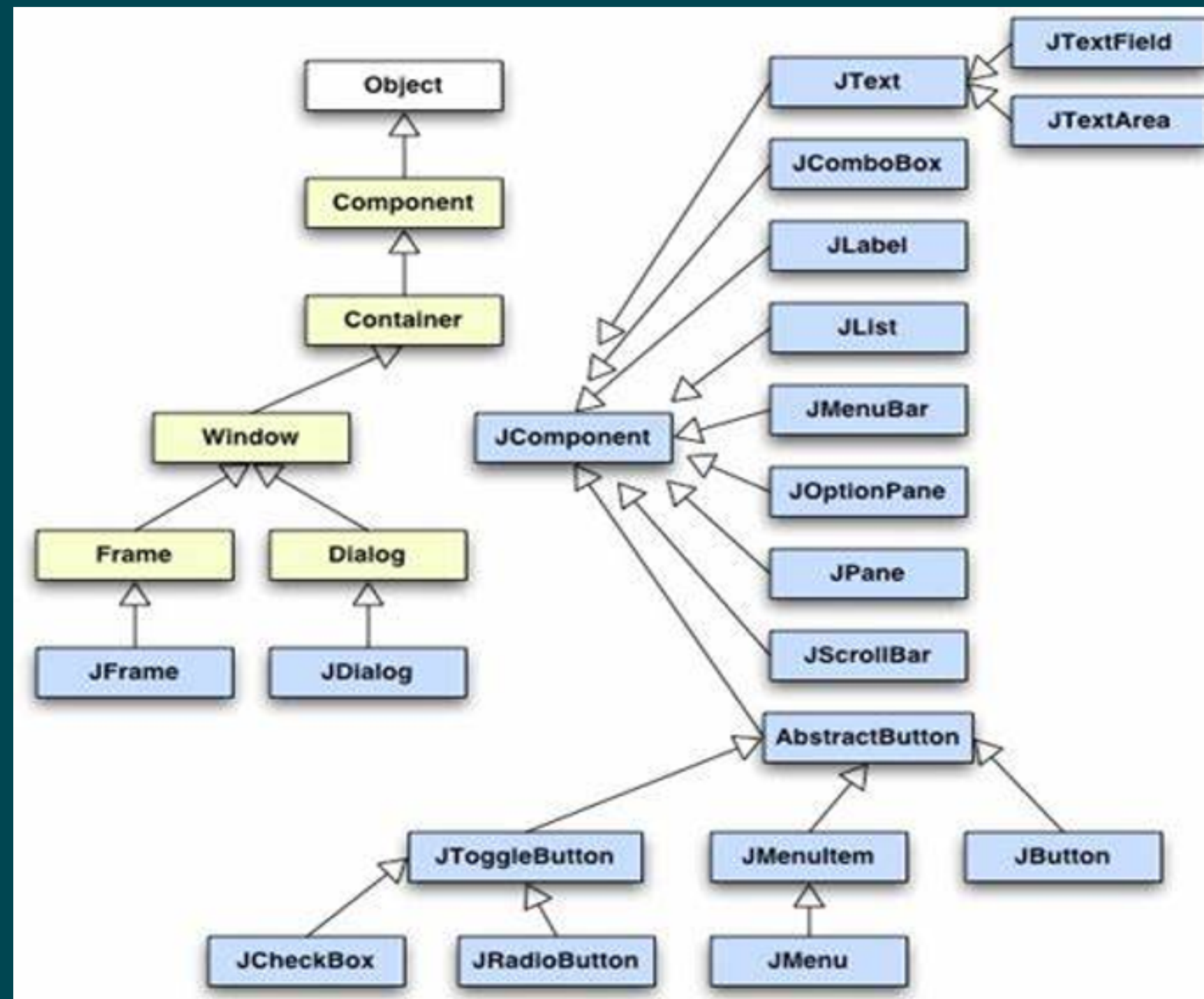
Esta classe traz as características comuns a todos os componentes, como parâmetros para suas dimensões e/ou métodos para manipular eventos . No caso do pacote swing a classe raiz é a **JComponent** (uma extensão da classe Component).

Uma subclasse da classe Component muito utilizada é a **Container**, que nada mais é do que um componente capaz de abrigar outros componentes.

Outras classes comuns são Frame e **JFrame** (pacote swing), uma extensão da classe Window.



Para melhor entendimento, essa é a hierarquia das classes apresentadas pelos pacotes:



Classes Painéis

As classes para painéis são utilizadas normalmente para conseguir uma **melhor organização e maior variedade no funcionamento dos frames**, além de permitirem um design melhor otimizado ao criá-los .

Alguns dos tipos de classe utilizadas nas GUI para criação de painéis são:

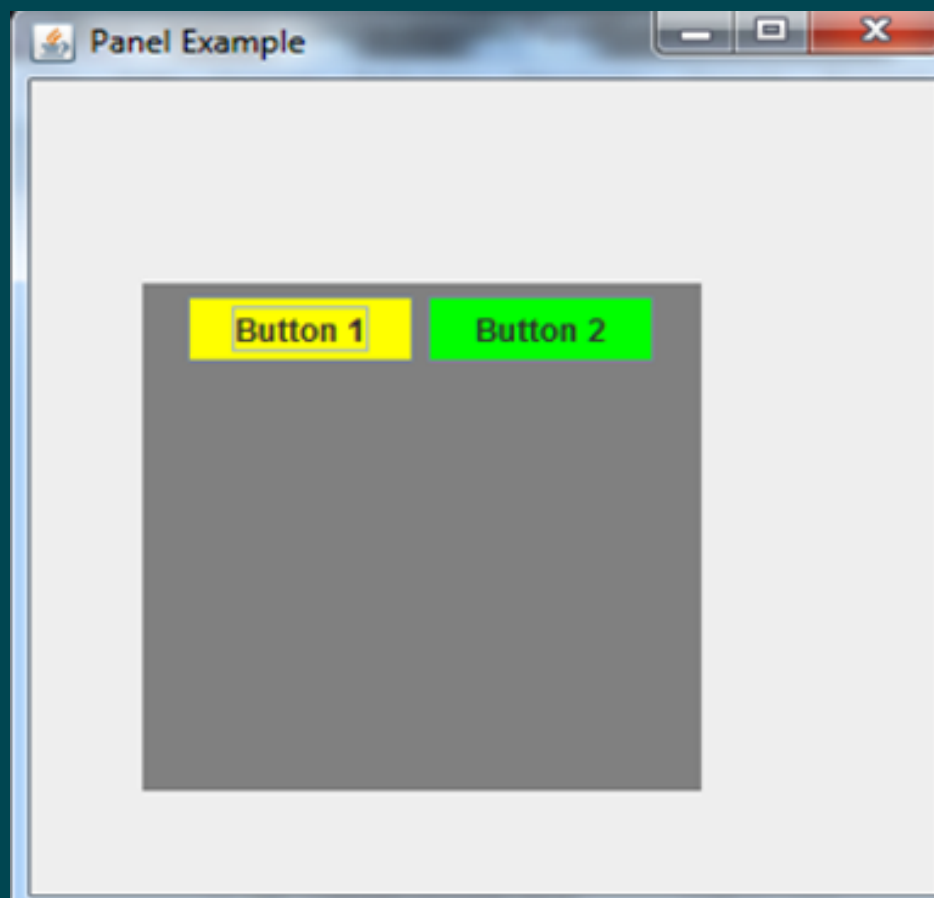
- JPanel
- JPanelEditor
- JPanelOption
- JPanelTabbed



PAINÉIS

JPanel

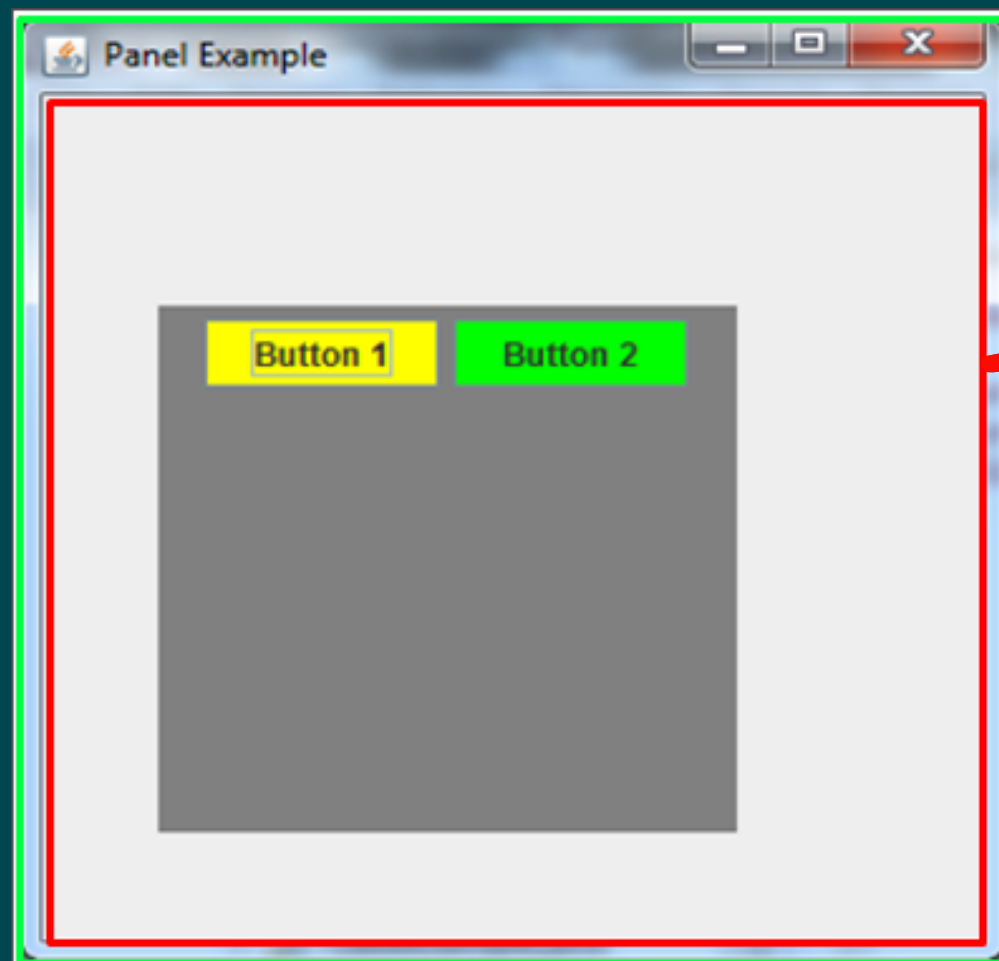
- Permite inserir qualquer componente da classe **JComponent** no painel, permitindo a interação com o usuário por meio de qualquer um deles.



- **Label:** Objeto que exibe um texto uma imagem ou ambos e não é editável.
- **TextField:** Recebe e exibe dados que o usuário digita e pode ser editável ou não.
- **ComboBox:** Cria uma lista de opções onde o usuário pode escolher uma delas.
- **JList:** Cria uma lista de opções onde o usuário pode escolher mais de uma.
- **Button:** Área que ao ser clicada aciona um evento.
- **CheckBox:** Caixa de seleção que pode ser marcada ou desmarcada.
- **JFrame:** Armazena os demais componentes.

JFrame

- JFrame representa a janela do programa com barra de título, ícone, botões de comando, etc.
- O painel vai ser encaixado no JFrame



JFrame

JPanel

Criação do JFrame

```
JFrame frame = new JFrame( title: "Exemplo1");  
frame.setContentPane(new Exemplo1().panel1);  
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
frame.pack();  
frame.setVisible(true);
```

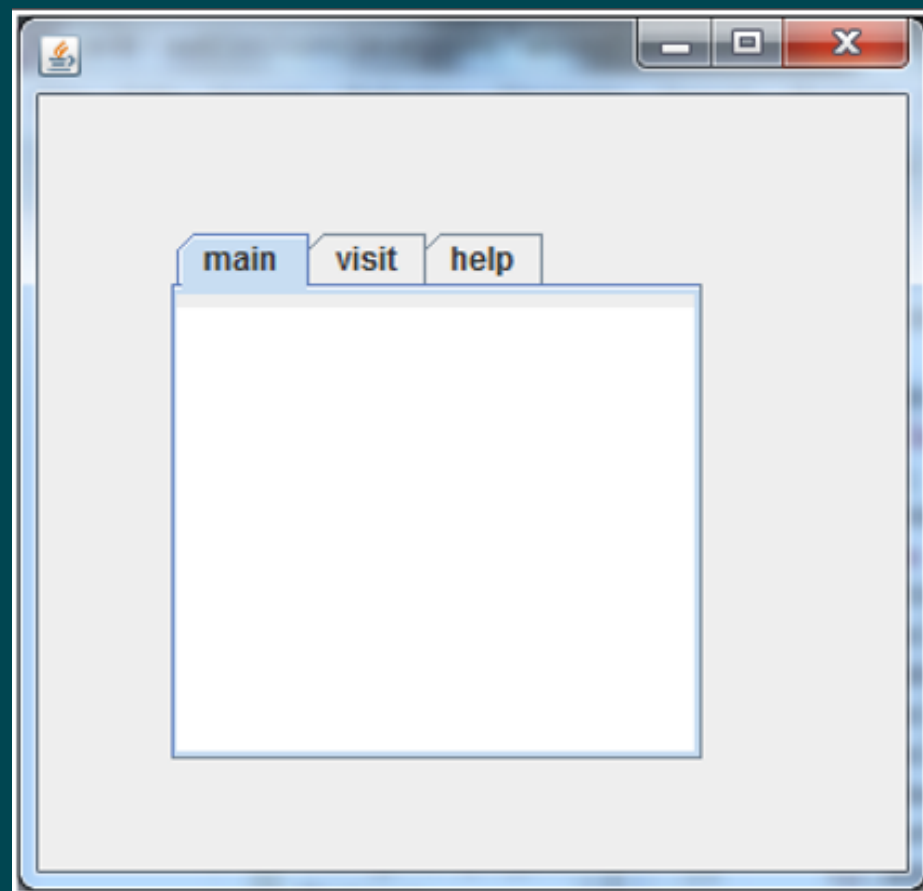
PAINÉIS

JTabbedPane

- Organiza os ícones dos componentes em abas com títulos específicos, permitindo que se alterne entre as abas e acesse seu conteúdo.



- Organiza melhor o conteúdo
- Evita a poluição visual
- Flexibilidade e rapidez ao trocar as guias



- Ao utilizar tais classes, podemos criar os mais diversos designs e funcionamentos para os frames presentes nos programas, e saber utilizá-las é uma verdadeira "mão na roda", pois assim, é possível criar **interfaces melhores e mais intuitivas** para os usuários no geral, podendo atender aos mais diversos objetivos.

Bibliografia

- <https://www.javatpoint.com>
- <https://docs.oracle.com>
- <https://www.javaprogressivo.net>
- <https://www.codejava.net>
- <https://www.devmedia.com.br>



```
If(apresentação  
== boa){  
    Nota = 10;  
}else{  
    Nota - -;  
}
```