

SCC0504 - Programação Orientada a Objetos

Interface Gráfica

Prof.: Leonardo Tórtoro Pereira

leonardop@usp.br

GUI

- Em Java, é possível criar interfaces de usuário com relativa simplicidade usando a biblioteca Swing
- A biblioteca JavaFX também contém funcionalidades para GUIs (Graphical User Interfaces), além da possibilidade de tocar mídias como filmes em mp4, áudios em mp3 entre outros

GUI

- Além disso, é possível criar gráficos e outras funcionalidades 2D com a biblioteca 2D ou a JavaFX
- E a JavaFX também permite importação de modelos 3D com arquivos OBJ e Maya

GUI

- Nesta aula, vamos focar em falar sobre interfaces gráficas com o uso de Swing
- É possível programá-las diretamente por código ou usar funcionalidades de IDEs como Netbeans, Eclipse e IntelliJ para tornar o trabalho mais fácil.
- Vamos ver os 2 métodos.

Java Swing

Java Swing

- Na biblioteca Swing existem 3 classes de container de alto nível que são de uso geral
 - ◆ JFrame
 - ◆ JDialog
 - ◆ JApplet

Java Swing

- É preciso se atentar a alguns fatos ao usar essas classes
 - ◆ Para aparecer na tela, todo componente de GUI precisa ser parte de uma hierarquia de contenção
 - É uma árvore de componentes que tem um container de alto-nível como raíz

Java Swing

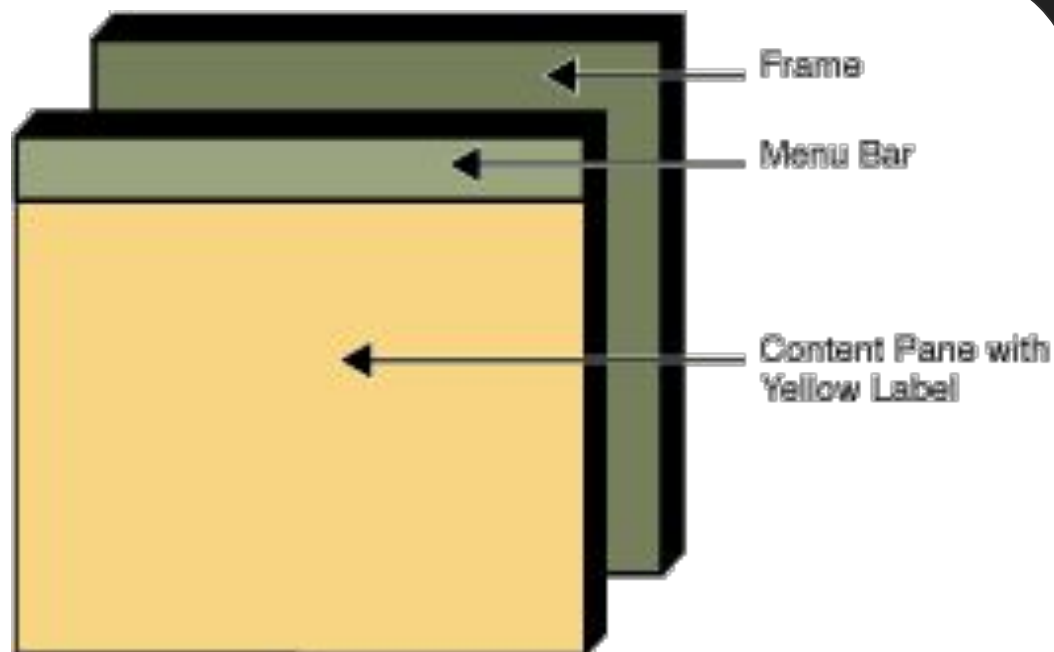
- É preciso se atentar a alguns fatos ao usar essas classes
- ◆ Cada componente de GUI pode ser contido apenas uma vez
 - Se ele já estiver em um container e você tentar adicioná-lo em outro, ele será removido do primeiro e adicionado ao segundo

Java Swing

- É preciso se atentar a alguns fatos ao usar essas classes
 - ◆ Cada container de alto nível tem um painel (*pane*) que, de maneira geral, contém (direta ou indiretamente) os componentes visíveis da GUI do container

Java Swing

- É preciso se atentar a alguns fatos ao usar essas classes
 - ◆ Pode-se, opcionalmente, adicionar uma barra de menu em um container de alto-nível, mas fora do *pane* de conteúdo



Fonte: [1]

Java Swing

- No exemplo anterior, temos um quadro (*frame*) criado por uma aplicação Java.
- O quadro contém uma barra de menu verde (com nenhuma opção de menu) e, no painel de conteúdo do quadro, uma *label* amarela, em branco.

```
public class TopLevelDemo {  
    private static void createAndShowGUI() {  
        //Create and set up the window.  
        JFrame frame = new JFrame("TopLevelDemo");  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        //Create the menu bar. Make it have a green background.  
        JMenuBar greenMenuBar = new JMenuBar();  
        greenMenuBar.setOpaque(true);  
        greenMenuBar.setBackground(new Color(154, 165, 127));  
        greenMenuBar.setPreferredSize(new Dimension(200, 20));  
        //Create a yellow label to put in the content pane.  
        JLabel yellowLabel = new JLabel();  
        yellowLabel.setOpaque(true);  
        yellowLabel.setBackground(new Color(248, 213, 131));  
        yellowLabel.setPreferredSize(new Dimension(200, 180));  
        //Set the menu bar and add the label to the content pane.  
        frame.setJMenuBar(greenMenuBar);  
        frame.getContentPane().add(yellowLabel, BorderLayout.CENTER);  
  
        //Display the window.  
        frame.pack();  
        frame.setVisible(true);  
    }  
}
```

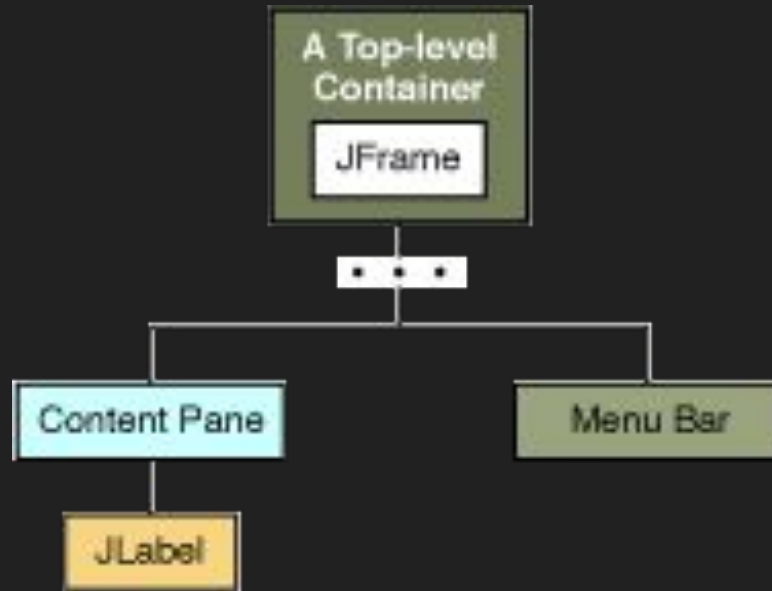
```
public static void main(String[] args) {  
    //Schedule a job for the event-dispatching thread:  
    //creating and showing this application's GUI.  
    javax.swing.SwingUtilities.invokeLater(new Runnable() {  
        public void run() {  
            createAndShowGUI();  
        }  
    });  
}
```

//Exemplo em

<https://docs.oracle.com/javase/tutorial/uiswing/examples/components/TopLevelDemoProject/src/components/TopLevelDemo.java>

Java Swing

→ O exemplo anterior continha a seguinte hierarquia:



Fonte: [1]

Java Swing

- Como regra, uma aplicação *standalone* com uma GUI baseada em Swing tem pelo menos uma hierarquia de contenção com um *JFrame* como raíz
- Se uma aplicação tem 1 janela principal e 2 diálogos, então a aplicação tem 3 hierarquias de contenção
 - ◆ E, portanto, 3 containers de alto nível
 - ◆ Uma com um *JFrame* como raíz
 - ◆ As outras duas com um *JDialog*

Java Swing

- Se for um *applet* baseado em Swing, possui pelo menos uma hierarquia de contenção
 - ◆ Exatamente uma tendo um *JApplet* como raíz
- Um *Applet* que apresenta um *dialog* tem 2 hierarquias de contenção
 - ◆ Os componentes no *browser* estão numa hierarquia com um *JApplet* como objeto
 - ◆ O diálogo tem um *JDialog* como raíz de sua hierarquia

Java Swing

- Para adicionar um componente a um painel é preciso encontrar o painel na hierarquia de alto nível chamando o método *getContentPane()*.
- ◆ O painel padrão é um container intermediário simples que herda de *JComponent*, e usa uma *BorderLayout* como gerente de *layout*
- No exemplo anterior:
 - ◆ `frame.getContentPane().add(yellowLabel, BorderLayout.CENTER);`

Java Swing

- O método *getContentPane* retorna um objeto *Container*, e não um *JComponent*
- Se você quiser usar algo de *JComponent*, é preciso dar um *cast* de tipo no retorno ou criar seu próprio componente para ser o painel de conteúdo
- ◆ Os exemplos em Java geralmente fazer o segundo, por ser mais “limpo”

Java Swing

- O gerente de *layout* padrão para um *JPanel* é um *FlowLayout*
 - ◆ Você provavelmente vai querer mudar.
- Para fazer de um componente o painel de conteúdo, use o método do container de alto nível *setContentPane*

Java Swing

```
//Create a panel and add components to it.
```

```
JPanel contentPane = new JPanel(new BorderLayout());
```

```
contentPane.setBorder(someBorder);
```

```
contentPane.add(someComponent, BorderLayout.CENTER);
```

```
contentPane.add(anotherComponent, BorderLayout.PAGE_END);
```

```
topLevelContainer.setContentPane(contentPane);
```

Java Swing

- O método *add()* e suas variantes *remove* e *setLayout* foram sobrecarregados para que as mudanças sejam passadas para o *contentPane* conforme for necessário
- Ou seja, para adicionar algo para o painel é possível escrever
 - ◆ *frame.add(child);*
- Mas isso só é válido para esses 3 métodos!

Java Swing

- Qualquer container de alto nível pode ter uma barra de menu
 - ◆ Mas ela é mais comum em *frames* e *applets*
- Para adicionar uma barra de menu é preciso
 - ◆ Criar um objeto *JMenuBar*
 - ◆ Populá-lo com menus
 - ◆ Chamar *setJMenuBar*

```
frame.setJMenuBar(greenMenuBar);
```

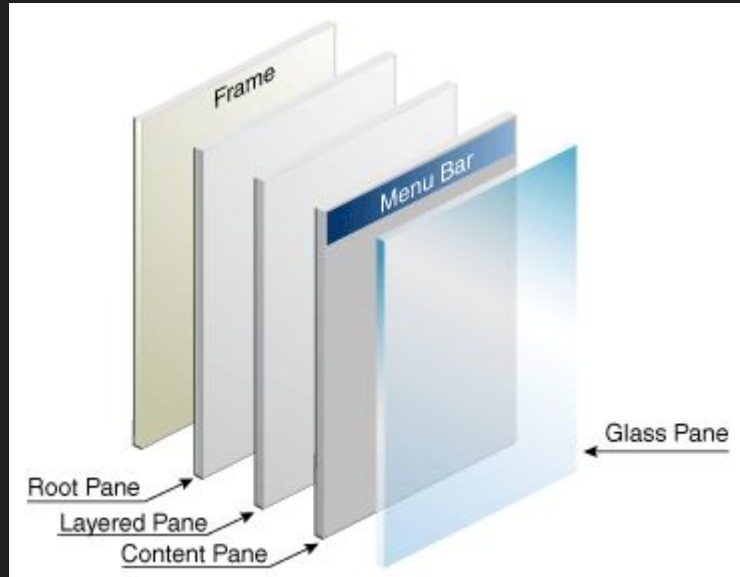
Java Swing

- Cada container de alto nível depende de um container intermediário chamado de *root pane*
 - ◆ Gerencia o conteúdo do painel e da barra de menu
 - ◆ Além de outros containers
- Geralmente não é preciso saber sobre ele para usar componentes. Mas para interceptar clicks de mouse ou pintar sobre múltiplos elementos, é preciso.

Java Swing

- Cada container de alto nível depende de um container intermediário chamado de *root pane*
 - ◆ Gerencia o conteúdo do painel e da barra de menu
 - ◆ Além de outros containers
- Geralmente não é preciso saber sobre ele para usar componentes. Mas para interceptar clicks de mouse ou pintar sobre múltiplos elementos, é preciso.

Java Swing



Fonte: [1]

Java Swing

- O *Layered Pane* contém a *menu bar* e o *content pane* e permite a ordenação em Z (*Z-ordering*) dos outros componentes
- O *Glass Pane* é normalmente usado para interceptar eventos de entrada ocorrendo sobre o container de alto nível
 - ◆ Também pode ser usado para pintar sobre múltiplos componentes

Java Swing

- Um dos jeitos mais fáceis de aprender a usar de fato a Swing é vendo exemplos.
- Neste link existem dezenas de exemplos
 - ◆ <https://docs.oracle.com/javase/tutorial/uiswing/examples/components/index.html#Converter>
- Vamos ver alguns agora

Java Swing

→ Tutoriais GUI IntelliJ

◆ <https://www.jetbrains.com/help/idea/designing-gui-major-steps.html>

→ Tutoriais GUI NetBeans

◆ https://netbeans.org/kb/docs/java/quickstart-gui_pt_BR.html

Referências

1. <https://docs.oracle.com/javase/tutorial/uiswing/components/toplevel.html>
2. <https://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm>
3. <https://docs.oracle.com/javase/8/javase-clienttechnologies.htm>
4. <https://docs.oracle.com/javase/tutorial/uiswing/index.html>
- 5.