

#### **Slide 42**

O JLabel, subclasse de JComponent, é um rótulo que identifica a finalidade dos componentes de uma GUI por meio ou de texto, imagem ou texto e imagem , muito útil em uma interface encorpada.

#### **Slide 43**

Num JFrame, que seu construtor utiliza um argumento String como texto na barra de título da janela, devesse encaixar os componentes nele, formando um layout. Há vários gerenciadores de layout, chamados IDE, para auxiliar nessa tarefa, também disponibilizados pelo Java.

#### **Slide 44**

Um IDE, no geral, simplifica a criação de GUI fornecendo ferramentas de design para modelar precisamente um layout.

#### **Slide 45**

O FlowLayout é uma classe que regula de forma simples como serão colocados os componentes dentro de um contêiner, alterar o formato do contêiner pode alterar o código da GUI. Os componentes ficam organizados da esquerda para a direita, se não houver mais espaço é colocado na linha de baixo.

#### **Slide 48**

Utilizando o método setToolTipText é possível, usando uma String, especificar o texto de um rótulo. É acionado ao posicionar do cursor do mouse de um usuário sobre os JComponent.

JComponent são adicionados ao contêiner utilizando o método add que é herdado indiretamente da classe Container.

#### **Slide 49**

Um ícone influencia na personalidade de uma GUI, sofisticando o layout. Uma imagem salva na pasta junto ao resto dos scripts java pode ser usada como argumento Icon para o um construtor ou para o método setIcon do componente.

**ImageIcon (pacote javax.swing) suporta vários formatos de imagem, incluindo Graphics Interchange Format (GIF), Portable Network Graphics (PNG) e Joint Photographic Experts Group (JPEG).**

#### **Slide 50**

**Para conseguir levar uma imagem até sua GUI é necessário invocar o comando `getClass().getResource("imagem.png")` . O `getClass` é usado para buscar uma referencia ao objeto, o `getResource` para encontrar a imagem e transformar a localização em URL, depois o `ImageIcon`, usando o URL, carrega a imagem na memoria da GUI.**

#### **Slide 51**

**Um `JLabel` pode tanto receber uma mensagem como um ícone ou os dois. Para posicionar eles é utilizado o `SwingConstants`, que tem a função de realocar os rótulos, aparecendo por padrão primeiro a imagem depois do texto quando um rótulo tem os dois. Os alinhamentos são configurados por `setHorizontalAlignment` e `setVerticalAlignment`, respectivamente.**

#### **Slide 53**

**A `JLabel` tem métodos para configurar a aparência de um rótulo assim que instanciado, como: o método `setText`, configura o texto exibido no rótulo; o método `getText`, recupera o texto atual exibido em um rótulo; o método `setIcon`, especifica o `Icon` a ser exibido em um rótulo; o método `getIcon`, recupera o `Icon` atual exibido em um rótulo; os métodos `setHorizontalTextPosition` e `setVerticalTextPosition`, especificam a posição do texto no rótulo.**

#### **Slide 54**

**Quando se fecha uma janela, ela só é ocultada. Para o programa terminar ao usuário fechar a janela usasse o método `setDefaultCloseOperation` com o argumento `JFrame.EXIT_ON_CLOSE`. Há outros métodos como `setSize` que especifica a largura e altura da janela em pixels e o método `setVisible` que com o argumento `true` exibe a janela na tela.**