

Resumo

Numa GUI extensa, temos dificuldade de identificar o objetivo de cada componente, para isso o tooltip declara a funcionalidade de cada componente.

Esse texto é conhecido como rótulo, ou Label. sendo criado com a classe JLabel usando uma subclasse de JComponent. É capaz de exibir texto e imagem ou ambos.

é usado uma String como texto na barra de título da janela

é necessário anexar cada componente GUI a um container, por exemplo JFrame.

geralmente é necessário posicionar elementos em posições específicas na tela. para isso o java nos fornece vários gerenciadores de layout.

vários IDEs fornecem ferramentas para fazer o design do programa, auxiliando o desenvolvedor, essas ferramentas fornecem opções diversas, tais como aumentar o tamanho dos botões e posicionar em um ponto específico da tela. depois de feito o design a IDE gera o código GUI automaticamente, o que simplifica muito o desenvolvimento.

FlowLayout

a ordem padrão de colocação dos componentes no container é da esquerda pra direita, e quando nao ha mais espaço da esquerda pra direita, eles são colocados na próxima linha. o que o FlowLayout faz é reposicionar esses componentes na tela de forma a adicionar ou remover linhas, consequentemente.

o método setLayout é herdado da classe Container, e deve ter em sua assinatura um objeto que implemente a interface Layout manager. ex FlowLayout.

se o componente não for adicionado ao gui explicitamente ele não será mostrado.

utilizar dicas de ferramentas ajuda o usuário a identificar a função do componente na tela.

o método setToolTip mostra uma string com alguma dica sobre aquele componente, quando o usuário coloca o mouse por cima.

para adicionar um componente a um container é utilizado o método add. que é herdado de Container.

ícones são utilizados para aprimorar a aparência ou especificar funcionalidade de algo. normalmente é especificado com o método setIcon. para ser implementado é necessário importar do pacote javax.swing. suporta várias extensões de imagens, como gif png jpeg.

`getClass().getResource("bug1.png")` carrega a imagem para ser usada no programa
essa pode ser carregada num `JLabel`. que pode receber um texto juntamente da
imagem.

geralmente o texto aparece a direita da imagem.

o `JLabel` pode ter os alinhamentos horizontais e verticais alterados pelos métodos
`setHorizontalAlignment` e `setVerticalAlignment`, respectivamente.

classe `JLabel` fornece métodos para alterar a aparência de um rótulo depois de ele
ter sido instanciado.

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 ícone a ser exibido em um rótulo.

O método `getIcon` recupera o ícone atual exibido em um rótulo.

Os métodos `setHorizontalTextPosition` e `setVerticalTextPosition` especificam a
posição do texto no rótulo.

Diferentemente do que pensamos, por padrão quando fechamos uma janela, ela é
somente o untada. Um modo de mudar isso é chamar o método
`setDefaultCloseOperation` com o
argumento `JFrame.EXIT_ON_CLOSE`.

O método `setSize` especifica a largura e altura da janela em pixels.

O método `setVisible` com o argumento `true` exibe a janela na tela.