

Objetivos: Aprender a usar diferentes tipos de diálogos ao exibir mensagens;  
Usar comentários para facilitar o entendimento do código;  
Disparar outros tipos de eventos;

Recursos: Aula em laboratório com uso do computador; Quadro e pincel; Roteiro de Aula;

## COMENTÁRIOS

Comentários são linhas de código não possuem qualquer função no programa senão a documentação do mesmo, a fim de facilitar o entendimento de quem lê o código. Observe o trecho abaixo:

**Arquivo:** Comentado.java

```
1. // isso não aparece ao executar
2. int x = 1 + 1;
3. /* este é um comentário
4.    de múltiplas
5.    linhas, ok?
6. */
```

Os caracteres //, /\* e \*/ são usados para delimitar os comentários. No NetBeans os comentários serão exibidos na cor cinza. É boa prática de programação colocar comentários antes de cada bloco de comando que possua funcionalidades importantes.

## EXIBINDO DIÁLOGOS PARA O USUÁRIO

Com o recurso de criação de formulários do NetBeans é possível criar diálogos personalizados para exibir mensagens para o usuário, entretanto existe uma forma bem simples de criar um diálogo. Através da classe javax.swing.JOptionPane é possível criar diálogos de mensagens, entrada de dados e confirmação. A seguir veremos como inserir esse recursos em nossos programas.

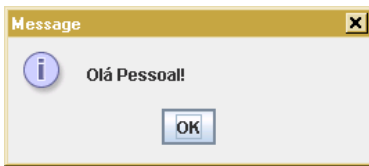
**Arquivo:** Dialogos.java

```
1. import javax.swing.JOptionPane;
2. // aqui continua o código gerado
3. // Dentro do código do botão escreva:
4. JOptionPane.showMessageDialog(null, "Olá Pessoal!");
```

O comando **import** deve ser usado na primeira linha do seu programa, ou logo após a declaração package (se houver), ele será usado para incluir no seu código as funcionalidades da classe indicada, nesse caso javax.swing.JOptionPane; Se você esquecer de fazer a importação, o NetBeans marca o comando com uma linha vermelha, e ao passar o mouse sobre essa linha ele exibe uma lâmpada do lado esquerdo, clique sobre ela para ver a sugestão de correção, então é só clicar para corrigir. Isso funciona para a maioria dos erros.



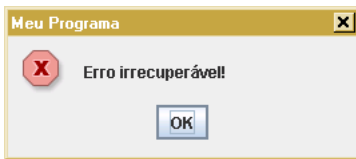
O código descrito anteriormente exibe na tela uma janela semelhante a esta:



Usamos dois parâmetros para formar essa janela: o termo **null** refere-se a um objeto que é responsável por esta mensagem, como não estamos usando nenhum objeto, null diz que a mensagem não possui um dono. Já o segundo parâmetro (que aparece entre aspas) é o texto que será exibido na tela.

Podemos ainda fazer outras modificações, como adicionar um título à janela e mudar o ícone exibido. Observe o código abaixo:

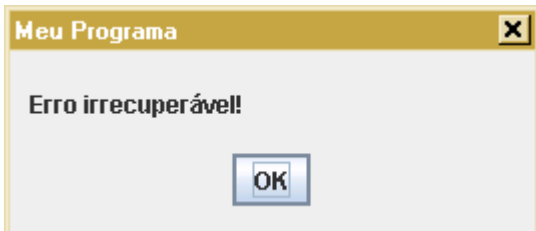
```
JOptionPane.showMessageDialog(null,"Erro irreversível!","Meu Programa",JOptionPane.ERROR_MESSAGE);
```



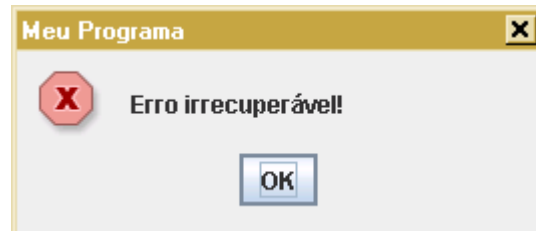
Observe que desta vez usamos 4 parâmetros: null e "Erro irreversível!" que são os mesmos explicados anteriormente, e dois novos: "Meu Programa" que é o título da janela, e JOptionPane.ERROR\_MESSAGE que modifica o ícone a ser exibido.

Os ícones possíveis são:

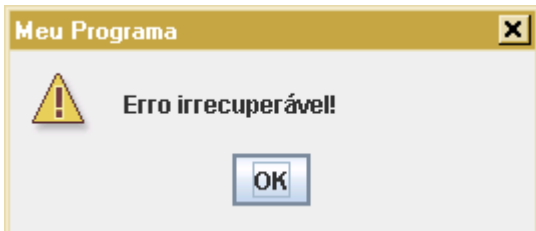
JOptionPane.PLAIN\_MESSAGE  
(Sem ícone)



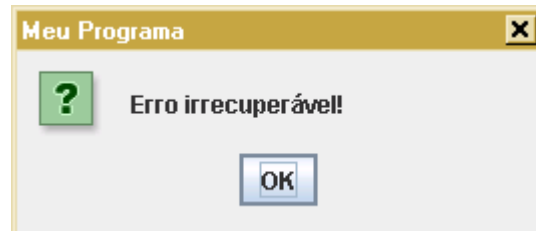
JOptionPane.ERROR\_MESSAGE  
(Para erros)



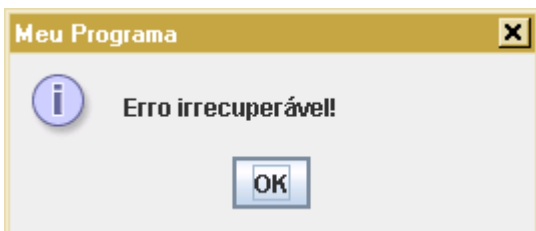
JOptionPane.WARNING\_MESSAGE  
(Atenção)



JOptionPane.QUESTION\_MESSAGE  
(Perguntas)



JOptionPane.INFORMATION\_MESSAGE  
(Informação)



Além de digitar a mensagem direto no código, é possível usar variáveis String no lugar do texto entre aspas, como no exemplo abaixo:

```
String msg = "Essa mensagem está numa variável";  
JOptionPane.showMessageDialog(null, msg);
```

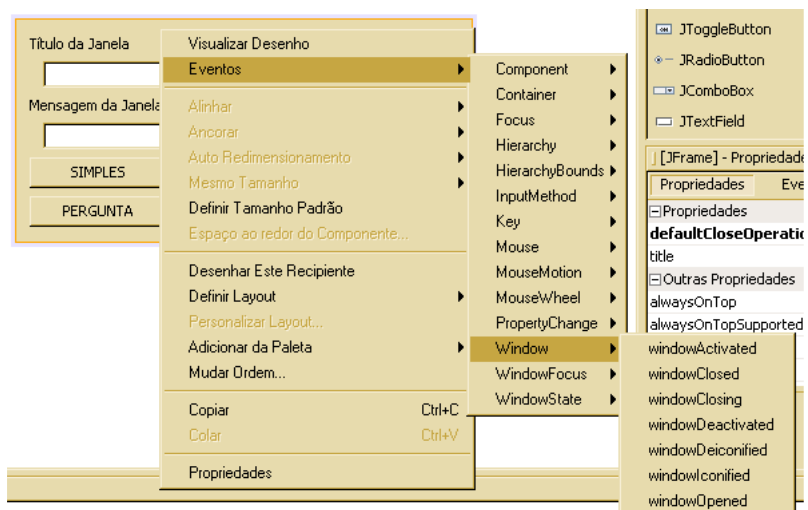
Dessa forma você pode montar a mensagem antes de exibi-la, ou mesmo combinar tipos diferentes de variáveis, fica a seu critério.

Se quiser exibir mensagens em múltiplas linhas, use o `\n` entre as palavras para fazer a quebra.

## EVENTOS DE JANELA

Até agora estamos usando eventos apenas nos botões para disparar os códigos que usamos. Vamos aprender outro tipo de evento que é aplicado às janelas.

Clique com o botão direito numa área vazia da janela, selecione **Eventos > Window**, uma lista de opções será exibida.

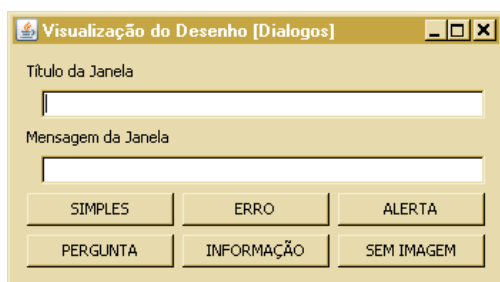


The screenshot shows a Java Swing window with a context menu open. The menu path is 'Eventos > Window'. The 'Window' sub-menu is expanded, showing a list of window events: windowActivated, windowClosed, windowClosing, windowDeactivated, windowDeiconified, windowIconified, and windowOpened. To the right of the screenshot, there are three text boxes explaining the use of these events:

- Use **windowClosing** para disparar um evento quando o X de fechar a janela for pressionado. Isso permite, por exemplo, confirmar se você quer fechar mesmo a janela.
- Use **windowOpened** para disparar um evento logo que a janela for exibida.
- Os demais eventos são para minimizar, restaurar, quando estiver ativa ou perder o foco. Teste cada um para ver como se comportam.

## EXERCÍCIOS

1. Crie um programa construtor de diálogos. Esse programa deve possuir um campo onde o usuário pode digitar uma mensagem, um segundo campo que recebe o título da janela e seis botões, que ao serem clicados, capturem os textos existentes nos campos e mostre o diálogo com a imagem correspondente. Coloque comentários de única linha e de múltiplas linhas que explique o código de cada botão. Coloque também uma mensagem tipo "obrigado por usar o programa" quando a janela for fechada. Veja o modelo:



Obs.: SIMPLES é a janela que só usa dois parâmetros (null e a mensagem).