

# Linguagens de Programação

Java Orientado a Objetos

Prof. Waldeck Lindoso Jr.

# Introdução a Interface GUI no Java

- Geralmente quando se está começando a programar, o desenvolvedor começa a fazer códigos que são retornados no console em formato de texto, pois muitos são códigos de aprendizagem. Mas quando é necessário desenvolver sistemas que precisam de alguma interação mais aprimorada com o usuário, utiliza-se as interfaces gráficas, então a ideia começa a evoluir, para ajudar nesse tipo de desenvolvimento usamos a interface GUI.



<terminated> Hello [Java Application] C:\Users\wlind\.p2\

Hello World!

# GUI Java

- Conhecida como Interface Gráfica com Usuário (Graphical User Interface – GUI), é onde os resultados são apresentados em modo gráfico.
- Essa interface é formada através de componentes GUI, conhecidos por controles ou widgets. Esses componentes são objetos que fazem a interação com usuário por teclado, mouse ou outros dispositivos que venham a servir para entrada de dados.



# Componentes

- Os componentes GUI Swing estão dentro do pacote **javax.swing** que são utilizados para construir as interfaces gráficas. Alguns componentes não são do tipo GUI Swing e sim componentes **AWT**. Antes de existir o GUI Swing, o Java tinha componentes **AWT** (Abstract Windows Toolkit) que faz parte do pacote **javax.awt**.




# Componentes (diferença **Swing** x **AWT**)

- A diferença entre o GUI Swing e AWT, é na aparência e comportamento dos componentes, ou seja, quando criado por AWT, a aparência e comportamento de seus componentes são diferentes para cada plataforma e enquanto feito por GUI Swing, a aparência e comportamento funcionam da mesma forma para todas as plataformas.
- Os componentes AWT são mais pesados, pois requerem uma interação direta com o sistema de janela local, podendo restringir na aparência e funcionalidade, ficando menos flexíveis do que os componentes GUI Swing.



# Componentes (**mais utilizados**)

- **JLabel** - Exibe texto não editável ou ícones.
  - **TextField** – Insere dados do teclado e serve também para exibição do texto editável ou não editável.
  - **Button** – Libera um evento quando o usuário clicar nele com o mouse.
  - **CheckBox** – Especifica uma opção que pode ser ou não selecionada.
  - **ComboBox** – Fornece uma lista de itens onde possibilita o usuário selecionar um item ou digitar para procurar.
  - **JList** – Lista de itens onde pode ser selecionado vários itens.
  - **Panel** – É a área onde abriga e organiza os componentes inseridos.
- 

# JOptionPane

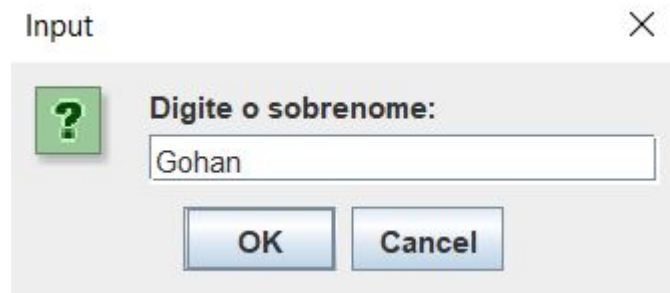
- Encontra-se no pacote **javax.swing**, e por ser um componente faz parte da Classe JComponent, permitindo ao usuário inserir informações nas caixas de diálogos, podendo exibir informações ou avisos.
  - exemplo de programa de leitura de nome e sobrenome.

```
String nome = JOptionPane.showInputDialog("Digite o nome: ");  
String sobreNome = JOptionPane.showInputDialog("Digite o sobrenome: ");  
String nomeCompleto = nome + " " + sobreNome;
```

```
JOptionPane.showMessageDialog(null, "Nome Completo: " + nomeCompleto,  
                             "Informação", JOptionPane.INFORMATION_MESSAGE);
```

# JOptionPane

- Saída do código anterior.





# JOptionPane

- Observe que os valores digitados na caixa de diálogo que aparece são automaticamente resultados das variáveis “nome” e “sobreNome” que estão armazenando dados conforme foi inserido através do método **showInputDialog**.
- A linha que invoca o método **showMessageDialog**, imprime os resultados armazenados na variável “**nomeCompleto**”. Abaixo a lista de argumentos desse método.



## JOptionPane (**argumentos**)

- Argumento que marca a posição que será exibida da caixa na tela, como não estamos trabalhando com frames o padrão é null.
- Argumento que insere na barra de título o que foi digitado no caso, “Informação” .
- O último argumento é o tipo da saída da mensagem que exhibe através do diálogo por meio de constante.



# JOptionPane (trabalhando com valores inteiros)

- Nesse caso é necessário fazer uma conversão das **Strings** para tipo de dados **int**, que são inseridas nas variáveis “**valorA**” e “**valorB**”, pois seus tipos são inteiros e caso seja inserido outro tipo, irá gerar um erro.

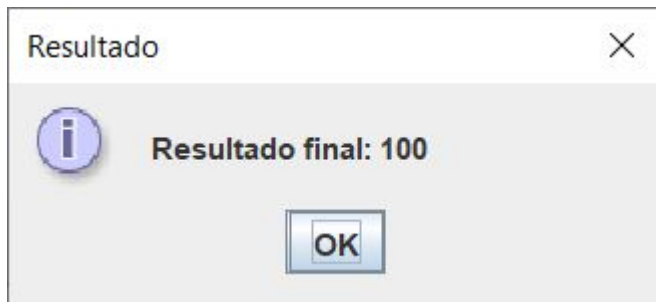
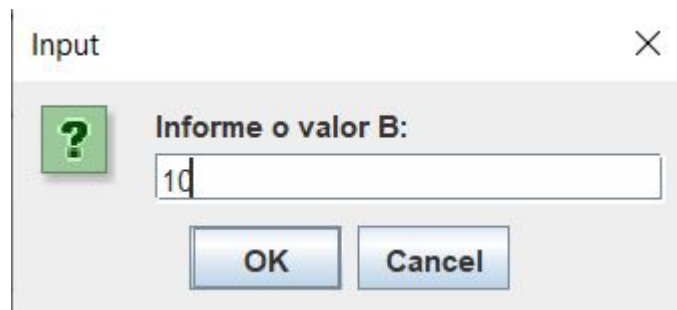
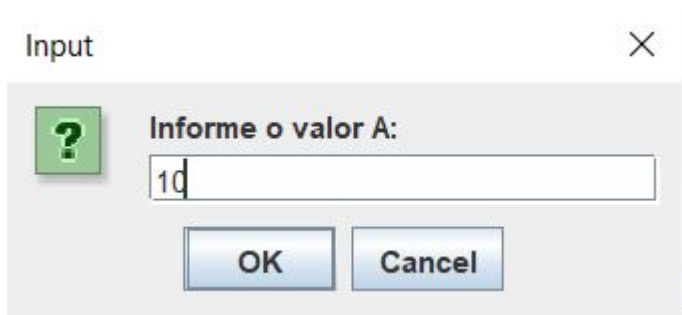
```
int valorA = Integer.parseInt(JOptionPane.showInputDialog("Informe o valor A: "));  
int valorB = Integer.parseInt(JOptionPane.showInputDialog("Informe o valor B: "));  
int resultado = valorA * valorB;
```

```
JOptionPane.showMessageDialog(null, "Resultado final: " + resultado,  
                             "Resultado", JOptionPane.INFORMATION_MESSAGE);
```



# JOptionPane

- Saída do código anterior.



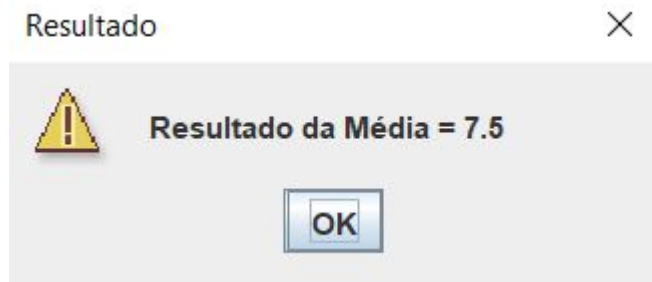
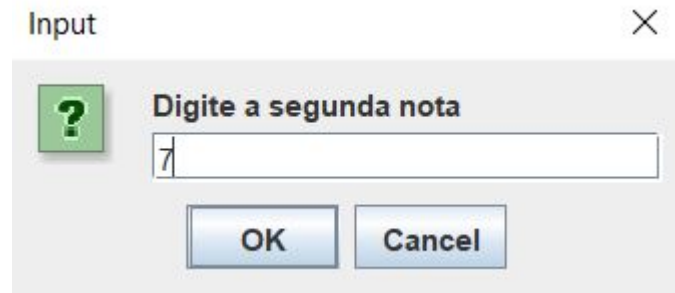
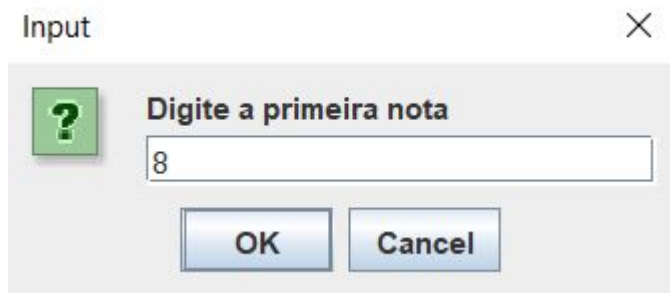
# JOptionPane (programa que calcula a média)

- Nesse caso é necessário fazer uma conversão das **Strings** para tipo de dados **float**, que são inseridas nas variáveis “**valorA**”, “**valorB**” e “**calcularMedia**”, pois seus tipos tem pontos flutuantes e caso seja inserido outro tipo, também irá gerar um erro.

```
float nota1, nota2, calculaMedia;  
nota1 = Float.parseFloat(JOptionPane.showInputDialog("Digite a primeira nota"));  
nota2 = Float.parseFloat(JOptionPane.showInputDialog("Digite a segunda nota"));  
calculaMedia = (nota1 + nota2) / 2;  
  
JOptionPane.showMessageDialog(null, "Resultado da Média = " + calculaMedia,  
                             "Resultado", JOptionPane.WARNING_MESSAGE);
```






# JOptionPane

- Saída do código anterior.



# Informação Diálogo de Mensagem

- Abaixo podemos observar as constantes que representam os tipos das mensagens quando são mostradas na saída de um resultado através **JOptionPane**.

	<b>INFORMATION_MESSAGE</b>	-> Indica uma mensagem Informativa
	<b>ERROR_MESSAGE</b>	-> Indica um erro ao usuário
	<b>QUESTION_MESSAGE</b>	-> Mostra uma questão ao usuário
	<b>WARNING_MESSAGE</b>	-> Alerta o usuário
	<b>PLAIN_MESSAGE</b>	-> Sem ícone

Obrigado!

