

# Programação de Computadores - I

Prof<sup>a</sup> Beatriz

Prof<sup>o</sup> Israel

A classe JOptionPane

# Uma introdução à Programação Visual

- ▶ A partir de agora abandonaremos um pouco o visual de console.
- ▶ Partiremos para a programação “quase” visual de Java.
- ▶ Quase visual porque por enquanto trabalharemos apenas com caixas de mensagem.
- ▶ Para isso falaremos um pouco sobre `JOptionPane` do pacote visual **Swing**.

# O que é a classe JOptionPane

- ▶ É uma classe que possibilita a criação de uma caixa de dialogo padrão que ou solicita um valor para o usuário ou retorna uma informação.
- ▶ Para utilizarmos a classe JOptionPane temos sempre que importar o pacote `javax.swing.JOptionPane` primeiro.
- ▶ Usamos então esta linha de comando: `import javax.swing.JOptionPane;`
- ▶ Sempre antes no inicio do código da classe.

# Métodos

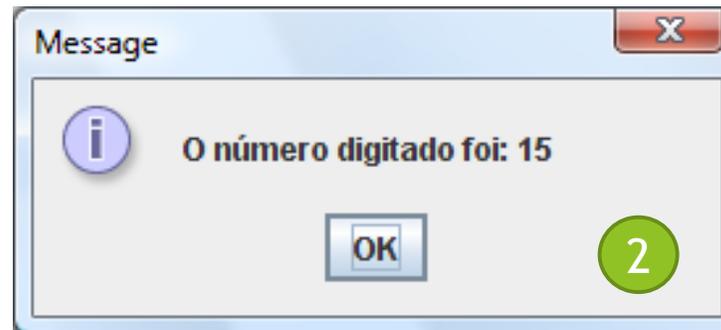
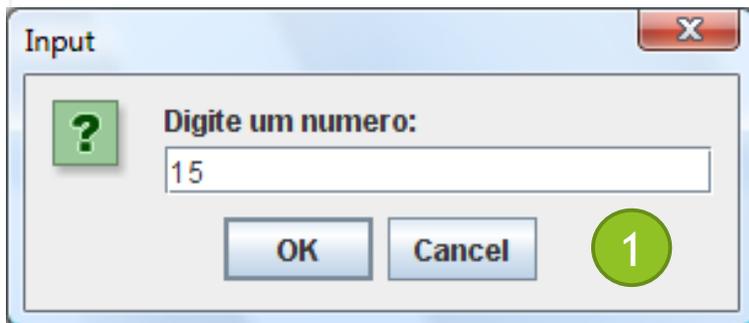
<b>Método</b>	<b>Descrição</b>	
showConfirmDialog	Solicita uma confirmação como(YES, NO, CANCEL)	
showInputDialog	Solicita algum valor	
showMessageDialog	Informa ao usuário sobre algo	
showOptionDialog	Unificação dos tres acim	

# Parâmetros

Parâmetro	Descrição
parentComponent	Define a caixa de diálogo onde irá aparecer todo o conteúdo. Há duas maneiras de definir a caixa de diálogo a primeira você mesmo cria utilizando os conceitos da classe JFrame. A segunda, você define esse parâmetro como null e o Java irá gerar uma caixa de diálogo padrão.
message	É a mensagem que o usuário deve ler. Esta mensagem pode ser uma simples String ou um conjunto de objetos.
messageType	Define o estilo da mensagem. O gerente de aparência pode expor a caixa de diálogo de formas diferentes, dependendo deste valor, pode fornecer um ícone padrão. Exemplos: <ul style="list-style-type: none"><li>•ERROR_MESSAGE</li><li>•INFORMATION_MESSAGE</li><li>•WARNING_MESSAGE</li><li>•QUESTION_MESSAGE</li><li>•PLAIN_MESSAGE</li></ul>
optionType	Define o conjunto de botões que irá aparecer na parte inferior da caixa de diálogo. Exemplos: <ul style="list-style-type: none"><li>•DEFAULT_OPTION</li><li>•YES_NO_OPTION</li><li>•YES_NO_CANCEL_OPTION</li><li>•OK_CANCEL_OPTION</li></ul>

# Demonstração rápida

```
import javax.swing.JOptionPane;  
// ou import javax.swing.*;  
  
public class TesteEntradaSaida {  
  
    public static void main(String[] args) {  
  
        String numero = JOptionPane.showInputDialog("Digite um numero:"); 1  
  
        2 JOptionPane.showMessageDialog(null, "O número digitado foi: " + numero);  
  
    }  
  
}
```



# MessageDialog

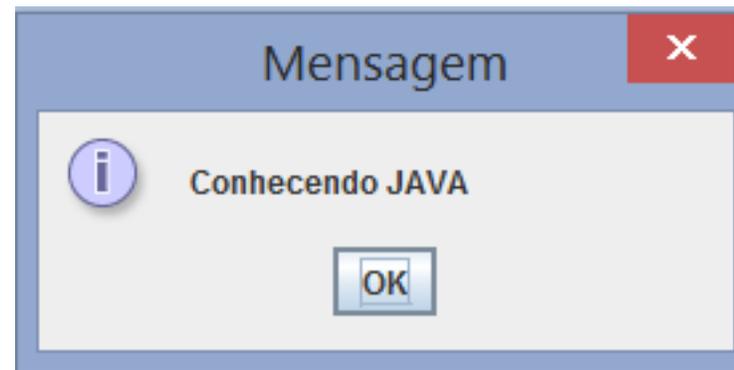
- ▶ Exibe uma caixa de diálogo com mensagens aos usuário.

## Código exemplo

```
import javax.swing.JOptionPane;

public class InoutDialog {
    public static void main(String[] args) {
        JOptionPane.showMessageDialog(null, "Conhecendo JAVA");
    }
}
```

## Resultado



# MessageDialog

## Código exemplo

```
import javax.swing.JOptionPane;
```

```
public class InoutDialog {  
    public static void main(String[] args) {  
        JOptionPane.showMessageDialog(null, "Conhecendo JAVA", "JAVA", JOptionPane.WARNING_MESSAGE);  
    }  
}
```

## Resultado



# ConfirmDialog

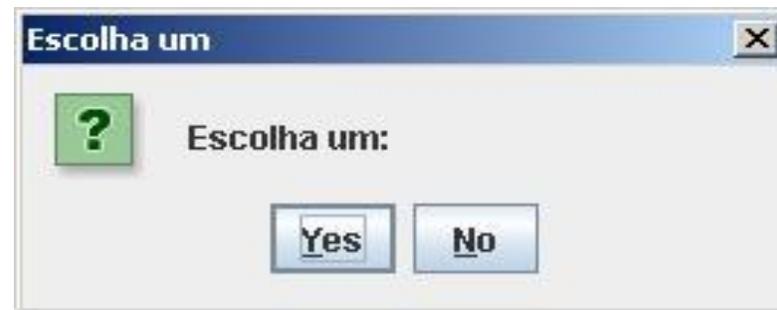
- Mostra um painel de informação com as opções Sim/Não e exibe a mensagem: 'Escolha um:'

## Código exemplo

```
import javax.swing.JOptionPane;

public class InoutDialog {
    public static void main(String[] args) {
        JOptionPane.showConfirmDialog(null, "Escolha um:",
            "Escolha um", JOptionPane.YES_NO_OPTION);
    }
}
```

## Resultado



# OptionDialog

- ▶ Mostra uma janela de aviso com as opções OK, CANCELAR, o texto 'Aviso' no título e a mensagem 'Clique em OK para continuar':

## Código exemplo

```
import javax.swing.JOptionPane;

public class InoutDialog {
    public static void main(String[] args) {
        Object[] options = { "OK", "CANCELAR" };
        JOptionPane.showOptionDialog(null, "Clique OK para
continuar", "Aviso",
        JOptionPane.DEFAULT_OPTION,
        JOptionPane.WARNING_MESSAGE,
        null, options, options[0]);}
}
```

## Resultado



# OptionDialog

- Note que se eu adicionar mais um item no vetor options, automaticamente adicionar mais um botão de opção com o nome que eu colocar.

## Código exemplo

```
import javax.swing.JOptionPane;

public class InoutDialog {
    public static void main(String[] args) {
        Object[] options = { "OK", "CANCELAR", "VOLTAR"
    };
    JOptionPane.showMessageDialog(null, "Clique OK para
continuar", "Aviso",
        JOptionPane.DEFAULT_OPTION,
        JOptionPane.WARNING_MESSAGE,
        null, options, options[0]);}
```

## Resultado



# InputDialog

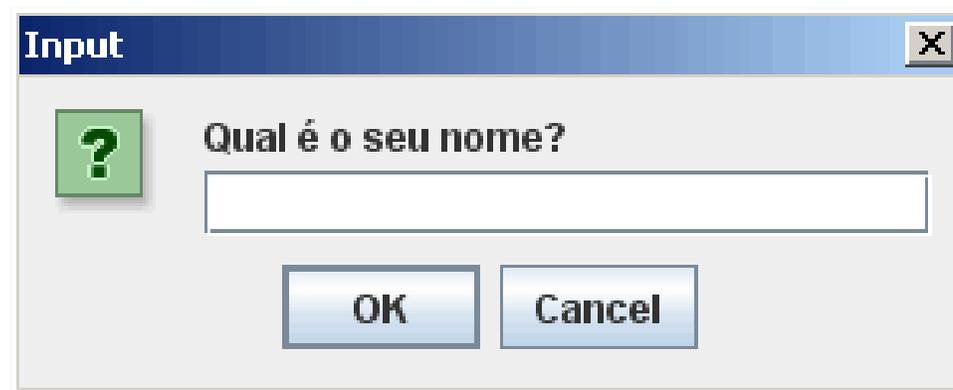
- ▶ Mostra uma caixa de diálogo solicitando que o usuário digite uma string

## Código exemplo

```
import javax.swing.JOptionPane;
```

```
public class InoutDialog {  
    public static void main(String[] args) {  
        String nome=JOptionPane.showInputDialog("Qual é o seu nome?");  
    }  
}
```

## Resultado



# InputDialog

- ▶ Mostra uma caixa de diálogo solicitando que o usuário selecione uma item.

## Código exemplo

```
import javax.swing.JOptionPane;

public class InoutDialog {
    public static void main(String[] args) {
        Object[] itens = { "MAÇA", "PERA", "BANANA" };
        Object selectedValue = JOptionPane.showInputDialog(null,
            "Escolha um item", "Opção",
            JOptionPane.INFORMATION_MESSAGE, null,
            itens, itens [0]); //}
    }
```

## Resultado

