

# Entrada e Saída de Dados

Profa. Karen Selbach Borges

# Entrada

# Classe Scanner

Introduzida a partir da versão 1.5 ou J2SE 5.

Implementa as operações de entrada de dados pelo teclado no console.

Para utilizar a classe Scanner deve-se proceder da seguinte maneira:

1. Importar `java.util. Scanner`;
2. Instanciar e criar um objeto Scanner: `Scanner ler = new Scanner(System.in);`
3. Ler valores através do teclado:
  - a. Para ler um valor inteiro: `int n = ler.nextInt();`
  - b. Para ler um valor real: `float preco = ler.nextFloat();` ou `double preco = ler.nextDouble();`
  - c. Para ler palavras simples que não usam o caractere de espaço: `String s = ler.next();`
  - d. Para ler palavras compostas, por exemplo, Pato Branco: `String s = ler.nextLine();`

# Como usar Scanner

Na leitura consecutiva de valores numéricos e String deve-se esvaziar o buffer do teclado antes da leitura do valor String, por exemplo:

```
System.out.printf("Informe um Número Inteiro: ");  
  
int n = ler.nextInt();
```

```
ler.nextLine(); // esvazia o buffer do teclado
```

```
System.out.printf("Informe uma cadeia de caracteres:\n");  
  
String s = ler.nextLine();
```

# Para ler apenas um caracter

É possível usar o método **read()** do pacote de classes **System.in**:

```
public static void main(String args[]) throws Exception {  
  
    System.out.printf("Informe um Caractere: ");  
  
    char c = (char)System.in.read();  
  
}
```

Saída

# Classe System.out

System.out representa a saída padrão de dados do sistema (console).

É possível utilizar os seguintes métodos:

- `println("String" + valor);`
  - observe que o + faz a concatenação da String com o valor, sendo que o resultante dessa operação é uma String.
- `printf("expressão_de_controle", argumento1, argumento2, ...);`
  - A expressão\_de\_controle deve ser uma sequência de caracteres (portanto, delimitada por aspas duplas) que determina as informações que serão mostradas na tela. Nesta expressão podem existir dois tipos de informações: caracteres comuns e códigos de controle (ou especificadores de formato).

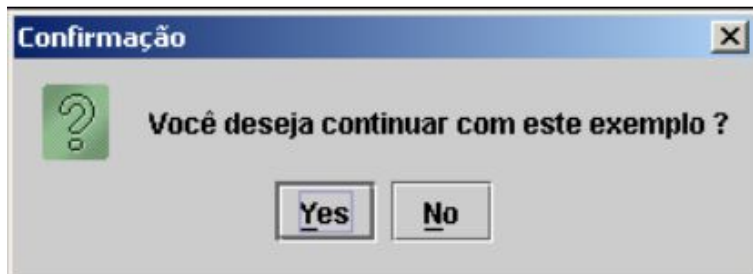
# Códigos de controle

Código	Formato (tipo de dados)
%c	Caractere simples (char)
%s	Cadeia de caracteres (String)
%d	Inteiro decimal com sinal (int)
%i	Inteiro decimal com sinal (int)
%ld	Inteiro decimal longo (long)
%f	Real em ponto flutuante (float ou double)
%e	Número real em notação científica com o "e" minúsculo (float ou double)
%E	Número real em notação científica com o "E" maiúsculo (float ou double)
%%	Imprimir o próprio caractere %

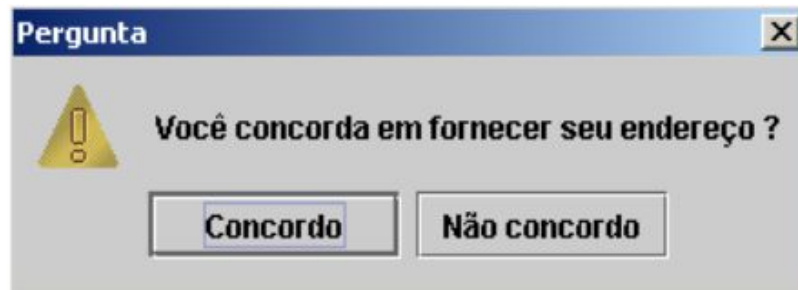


# Caixas de Diálogo

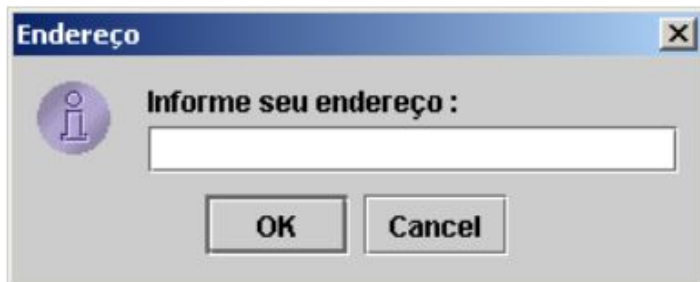
# Usando caixas de Diálogo



ConfirmDialog



OptionDialog



InputDialog



MessageDialog

# Classe JOptionPane

Existe um método específico para a criação de cada um dos tipos de caixas de diálogo.

- `showConfirmDialog`: solicita uma confirmação como(YES, NO, CANCEL)
- `showInputDialog`: solicita entrada de valores
- `showMessageDialog`: informa ao usuário sobre algo
- `showOptionDialog`: configurável.

Esses métodos possuem parâmetros que permitem realizar um conjunto de configurações.

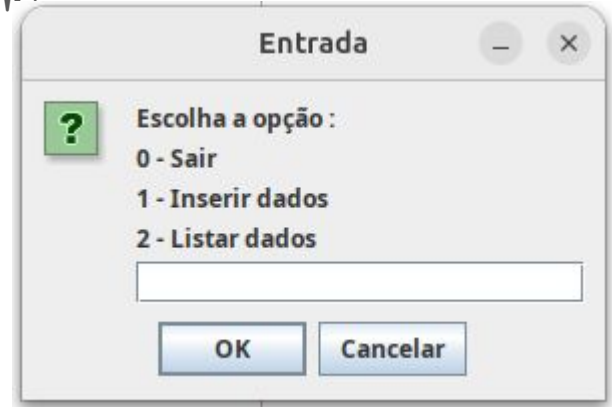
# Classe JOptionPane

Parametro	Descrição	
parentComponent	Define a caixa de diálogo onde irá aparece 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 parametro 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 dialogo de formas diferentes, dependendo deste valor, pode fornecer um icone 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>	

# InputDialog

```
import javax.swing.*;
public class ShowInputDialogExample {

    public static void main(String[] args) {
        String s = JOptionPane.showInputDialog(null, "Escolha a
        opção : \n0 - Sair "
            + "\n1 - Inserir dados"
            + "\n2 - Listar dados");
        int n = Integer.parseInt(s);
    }
}
```

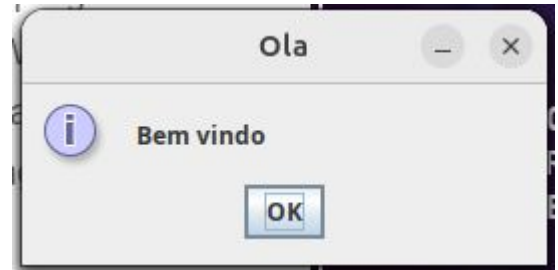


# MessageDialog

```
import javax.swing.*;

public class ShowMessageDialogExample {

    public static void main(String[] args) {
        JOptionPane.showMessageDialog(null, "Bem vindo", "Ola",
        JOptionPane.INFORMATION_MESSAGE);
    }
}
```

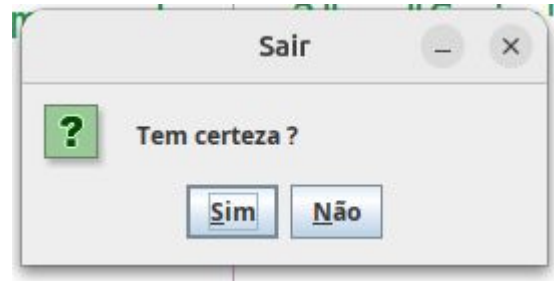


# ConfirmDialog

```
import javax.swing.*;

public class ShowConfirmDialogExample {

    public static void main(String[] args) {
        int opcao = JOptionPane.showConfirmDialog(null, "Tem
certeza ?", "Sair", JOptionPane.YES_NO_OPTION );
    }
}
```

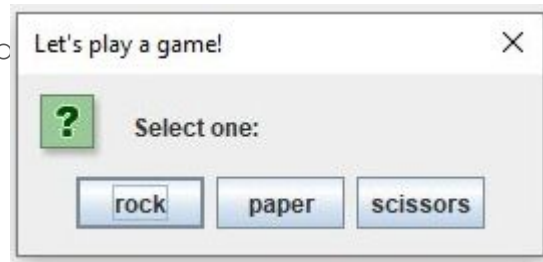


# OptionDialog

```
import javax.swing.*;

public class ShowOptionDialogExample {

    public static void main(String[] args) {
        String[] options = { "rock", "paper", "scissors" };
        var selection = JOptionPane.showOptionDialog(null, "Select one:", "Let's play
a game!", 0, 3, null, options, options[0]);
        if (selection == 0) {
            JOptionPane.showMessageDialog(null, "You chose rock!");
        }
        if (selection == 1) {
            JOptionPane.showMessageDialog(null, "You chose paper.");
        }
        if (selection == 2) {
            JOptionPane.showMessageDialog(null, "You chose scissors.");
        }
    }
}
```





# Referências

Entrada de Dados: Classe Scanner. Blog da DevMedia. Disponível em

<https://www.devmedia.com.br/entrada-de-dados-classe-scanner/21366>

Como funciona a classe Scanner do Java? Blog da DevMedia. Disponível em

<https://www.devmedia.com.br/como-funciona-a-classe-scanner-do-java/28448>

Javax.Swing.JOptionPane – Conhecendo e utilizando a classe JOptionPane. Bruno Augusto. Disponível em

<https://brunoagt.wordpress.com/2011/03/28/javax-swing-joptionpane-conhecendo-e-utilizando-a-classe-joptionpane/>

Java's JOptionPane showDialog by Example. Cameron McKenzie. The Server Sides' Blog. Disponível em

<https://www.theserverside.com/blog/Coffee-Talk-Java-News-Stories-and-Opinions/Javas-JOptionPane-showOptionDialog-by-Example>