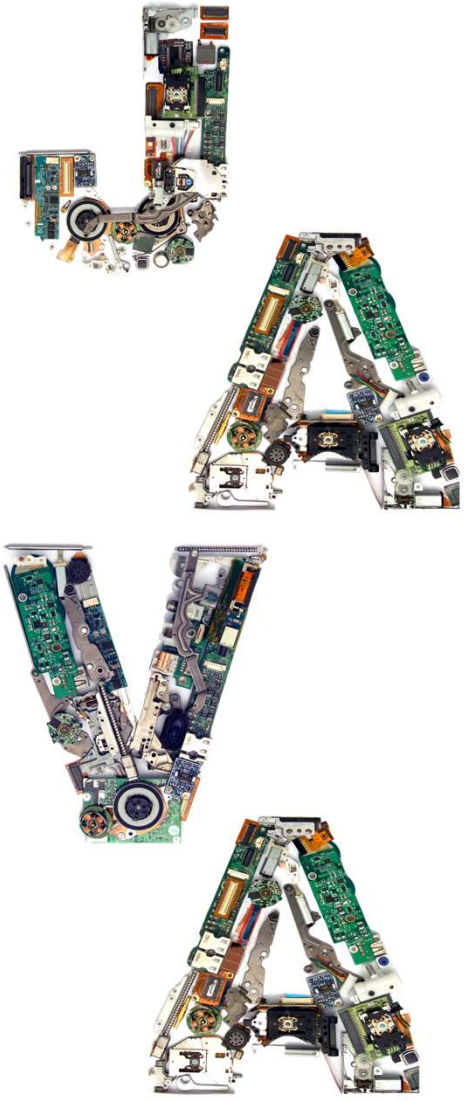
Programação Orientada a Objetos com Java e WEB

Janelas Gráficas



Introdução

A linguagem de programação Java apresenta algumas caixas de diálogos (janelas gráficas) para interação com o usuário do sistema.

A classe ***JOptionPane*** proporciona uma série de métodos estáticos que ao serem invocados criam caixas de diálogos simples e objetivas.

Para usar ***JOptionPane*** temos sempre que importar o pacote ***javax.swing.JOptionPane*** primeiro.

A classe ***JOptionPane*** apresenta caixas de diálogo para emitir uma simples mensagem no vídeo ou para fazer entrada de dados.

Caixa de diálogo para saída de dados (mensagem)

A caixa de diálogo de mensagem é uma caixa que serve apenas para emitir uma mensagem. Esta caixa também é configurável e versátil, pois serve para muitas situações distintas como uma mensagem de erro, um alerta, ou simplesmente uma informação.

O método ***showMessageDialog()*** é responsável em trazer a caixa de diálogo. Esse método pode receber vários argumentos.

Caixa de diálogo para saída de dados (mensagem)

Método *showMessageDialog()* com dois argumentos

Importação da classe *JOptionPane* do pacote *javax.swing*

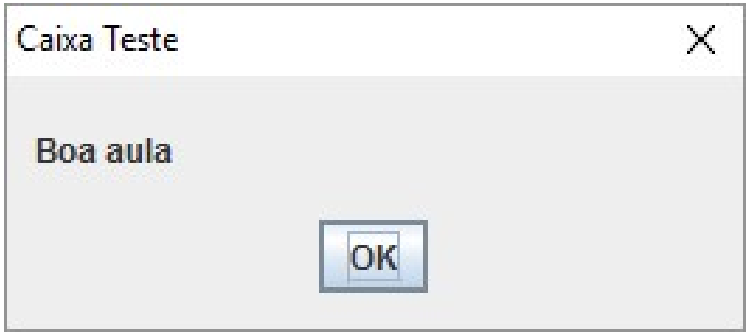
```
import javax.swing.JOptionPane;

public class Mensagem {
   public static void main(String[] args) {
      JOptionPane.showMessageDialog(null, "Boa aula");
   }
}
```

Alinhamento da janela → *null* indica que a janela será centralizada no vídeo.

Mensagem a ser exibida para o usuário

Caixa de diálogo para saída de dados (mensagem)

Método *showMessageDialog()* com quatro argumentos



alinhamento

título da janela

```
showMessageDialog(null, "Boa aula", "Caixa Teste", JOptionPane.DEFAULT_OPTION);
```

mensagem para o usuário

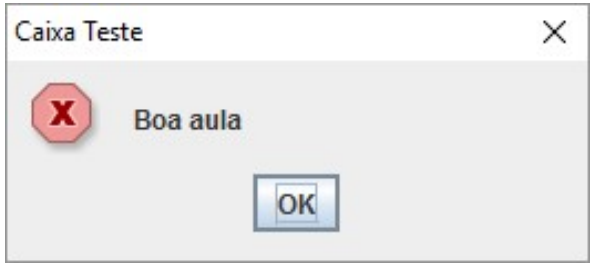
tipo da mensagem

Caixa de diálogo para saída de dados (mensagem)

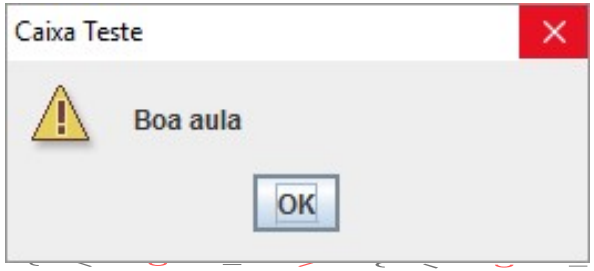
Tipos de mensagens *showMessageDialog()*

`JOptionPane.`***`DEFAULT_OPTION`***

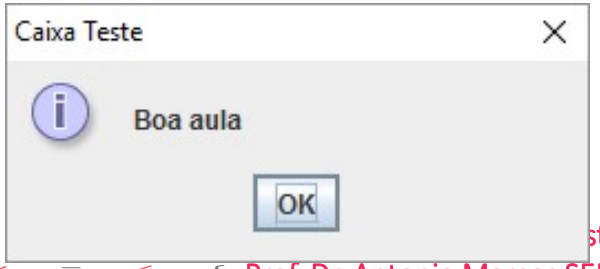
`JOptionPane.`***`ERROR_MESSAGE`***



`JOptionPane.`***`WARNING_MESSAGE`***



`JOptionPane.`***`INFORMATION_MESSAGE`***



Caixa de diálogo de confirmação

Outra caixa de diálogo simples e objetiva do ***JOptionPane*** é a caixa de diálogo de confirmação ou ***Confirm Dialog.***

A ***Confirm Dialog*** (caixa de confirmação) consiste de uma caixa contendo uma mensagem, um ícone e três botões: sim, não e cancelar.

Apesar deste ser o aspecto padrão, esta caixa, como qualquer outra de ***JOptionPane***, pode ser facilmente configurada (assunto que será tratado com mais detalhes nas próximas páginas).

O método ***showConfirmDialog*** sempre retorna uma constante que é a resposta clicada pelo usuário.

Caixa de diálogo de confirmação

Método *showConfirmDialog()* com dois argumentos

Importação da classe *JOptionPane* do pacote *javax.swing*

```
import javax.swing.JOptionPane;

public class Mensagem {
    public static void main(String[] args) {
        int resp;
        resp = JOptionPane.showConfirmDialog(null, "Deseja finalizar?");
    }
}
```

Retorno do método

Alinhamento da janela → *null* indica que a janela será centralizada no vídeo.

Mensagem a ser exibida para o usuário

Caixa de diálogo de confirmação

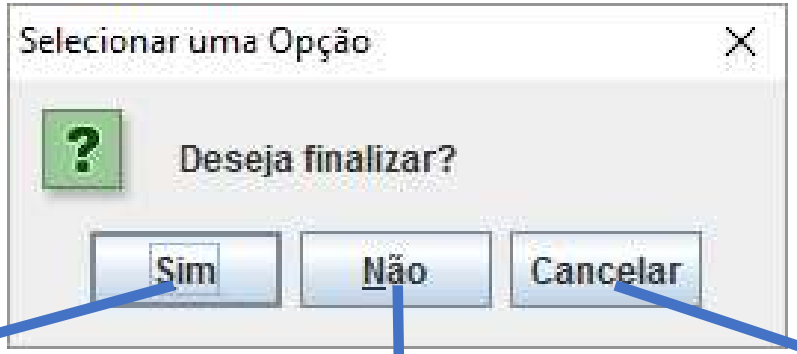
Método *showConfirmDialog()* com dois argumentos

```
import javax.swing.JOptionPane;

public class Mensagem {
    public static void main(String[] args) {
        int resp;
        resp = JOptionPane.showConfirmDialog(null, "Deseja finalizar?");
        if(resp == JOptionPane.YES_OPTION) {
            JOptionPane.showMessageDialog(null, "Programa será finalizado");
        }
    }
}
```

Caixa de diálogo de confirmação

Método *showConfirmDialog()* com dois argumentos

```
resp = JOptionPane.showConfirmDialog(null, "Deseja finalizar?");
```



`JOptionPane.YES_OPTION` Número → 0

`JOptionPane.NO_OPTION` Número → 1

`JOptionPane.CANCEL_OPTION` Número → 2

Caixa de diálogo para a entrada de dados

As caixas de diálogo de entrada de texto ou ***Input Text Dialog*** servem para fazer uma requisição de algum dado ao usuário de forma bem simples e direta.

O que é digitado pelo usuário é retornado pelo método em forma de ***string.***

Existem mais de 10 métodos sobrecarregados para invocar uma caixa de diálogo ***Input Text***, mas, a princípio, usaremos a mais simples.

O método ***showInputdDialog()*** recebe um argumento que é a string contendo a informação desejada, o que na maioria das vezes é uma pergunta ou pedido.

Caixa de diálogo para a entrada de dados

Método *showInputDialog()* com um argumento

Importação da classe *JOptionPane* do pacote *javax.swing*

```
import javax.swing.JOptionPane;

public class Mensagem {
   public static void main(String[] args) {
      String nome;
      nome = JOptionPane.showInputDialog("Informe seu nome");
   }
}
```

variável do tipo String para armazenar a entrada de dados

Mensagem a ser exibida para o usuário

Caixa de diálogo para a entrada de dados

Método *showInputDialog()* com um argumento

```
import javax.swing.JOptionPane;

public class Mensagem {
   public static void main(String[] args) {
       String auxIdade;
       int idade;
       auxIdade = JOptionPane.showInputDialog("Qual sua idade?");
       idade = Integer.parseInt(auxIdade);
   }
}
```

Método estático *parseInt()* da classe *Integer* converte uma *String* para inteiro

Caixa de diálogo para a entrada de dados

Método *showInputDialog()* com um argumento

```
import javax.swing.JOptionPane;

public class Mensagem {
    public static void main(String[] args) {
        String auxNota;
        double nota;
        auxNota = JOptionPane.showInputDialog("Qual sua nota?");
        nota = Double.parseDouble(auxNota);
    }
}
```

Método estático *parseDouble()* da classe *Double* converte uma *String* para double

Importação de membros estáticos

Métodos estáticos devem ser referidos usando o nome da classe à qual o membro estático pertence, por exemplo: ***Math.sqrt()***;

É possível usar métodos estáticos sem a referência da classe. Para isso deve ser usado o comando ***import*** estático;

Exemplo:

```
import static javax.swing.JOptionPane.*;
```

Importação de membros estáticos

Importação *static*. Detalhe para o asterisco (*) no final da linha.

```
import static javax.swing.JOptionPane.*;
import static java.lang.Double.*;

public class Mensagem {
   public static void main(String[] args) {
      String auxNota;
      double nota;
      auxNota = showInputDialog("Qual sua nota?");
      nota = parseDouble(auxNota);
   }
}
```

Usa-se apenas os nomes dos métodos, sem usar o nome das classes.

Conversão de tipos

Conversão de String para um valor numérico

```
public static void main(String[] args) {
    String s = "123";
    int i = Integer.parseInt(s);
    double d = Double.parseDouble(s);
    long l = Long.parseLong(s);
    float f = Float.parseFloat(s);
}
```

A conversão é feita utilizando o método *static* parseXXX().

Conversão de tipos

Conversão de um valor numérico para String

```
public static void main(String[] args) {
    int i = 123;
    double d = 123.123;
    long l = 123;
    float f = 123.2f;

    String s1 = String.valueOf(i);
    String s2 = String.valueOf(d);
    String s3 = String.valueOf(l);
    String s4 = String.valueOf(f);
}
```

A conversão é feita utilizando o método *static* sobrecarregado *valueOf*().

Bibliografia

- DEITEL, H. M., DEITEL, P. J. **JAVA como programar**. 10ª edição. São Paulo: Prentice-Hall, 2010.
- SCHILDT, H. **Java para Iniciantes – Crie, Compile e Execute Programas Java Rapidamente**. 6ª Edição, Editora Bookman, Porto Alegre, RS, 2015.