

Entrada e Saída de dados por JOptionPane

As caixas de diálogo têm uma função importante na interação do programa com o usuário. É através delas que dados são solicitados pelo programa e é através delas que mensagens de erro, informações, alertas e avisos podem ser exibidos para o usuário.

A classe `javax.swing.JOptionPane` facilita a tarefa de exibir diálogos padronizados que solicitem algum valor ao usuário ou que exibam alguma informação. Apesar da aparente complexidade, em função do grande número de métodos que ela implementa, a maioria das tarefas realizadas com essa classe se resumirá a chamadas de uma única linha a um dos seguintes métodos.

- `showMessageDialog()`; Caixa de diálogo que apresenta uma mensagem, possibilitando acrescentar ícones de alerta ao usuário;
- `showConfirmDialog()`; Caixa de diálogo que, além de emitir uma mensagem, possibilita ao usuário responder a uma pergunta;
- `showInputDialog()`; Caixa de diálogo que, além de emitir uma mensagem, permite a entrada de um texto;
- `showOptionDialog()`; Caixa de diálogo que abrange os três tipos anteriores.

Uma característica particular da classe `JOptionPane` é a não necessidade de declaração de objetos para a criação das caixas, que são criadas diretamente pelos métodos dessas classes.

`showMessageDialog()`





A caixa de diálogo `showMessageDialog()` é utilizada para mostrar ao usuário alguma informação e não retorna nenhum tipo de valor de resposta. Essas caixas de diálogo podem apresentar um ícone referente ao tipo de mensagem (pergunta, informação, alerta, ou erro) ou não apresentar nada, apenas a mensagem na tela sem ícone.

Sintaxe:

```
JOptionPane.showMessageDialog(Component, <mensagem>, <título da mensagem>, <tipo de mensagem>)
```

Onde:

- **Component:** Refere-se a um objeto do tipo contêiner que permite definir a posição da tela em que a caixa de mensagem aparecerá. Normalmente esse argumento é deixado como "null" (default) para que a mensagem apareça centralizada na tela.
- **Mensagem:** É a mensagem do tipo String que a caixa de diálogo informa ao usuário.
- **Título da mensagem:** É o texto que aparece na barra de título da janela da caixa de diálogo.
- **Tipo da mensagem:** É o ícone que representa o tipo de mensagem que está sendo informado ao usuário. Esse ícone é definido pelo usuário. A tabela abaixo apresenta as "constantes" da linguagem Java para definição dos ícones a serem apresentados na tela.

Ícone	Comando
 Pergunta	JOptionPane.QUESTION_MESSAGE
 Informação	JOptionPane.INFORMATION_MESSAGE
 Alerta	JOptionPane.WARNING_MESSAGE
 Erro	JOptionPane.ERROR_MESSAGE
Vazio (somente mensagem)	JOptionPane.PLAIN_MESSAGE

showConfirmDialog()

As caixas de diálogo de confirmação são pequenas janelas que possibilitam ao usuário responder algum questionamento por meio dos botões "YES, NO e CANCEL". Uma vez apresentada na tela, o usuário escolhe uma das opções e, dependendo do botão clicado, é gerado um valor inteiro pertencente à classe JOptionPane:

- YES_OPTION = 0;
- NO_OPTION = 1;
- CANCEL_OPTION = 2.

Estes valores são utilizados para conhecer qual dos botões foi clicado pelo usuário.

Existem três tipos de caixa de diálogo de confirmação:

- Apresentam os botões YES e NO → **YES_NO_OPTION** ou 0;
- Apresentam os botões YES, NO e CANCEL → **YES_NO_CANCEL** ou 1;
- Apresentam os botões OK e CANCEL → **OK_CANCEL_OPTION** ou 2.

Sintaxe:

```
JOptionPane.showConfirmDialog(Component, <mensagem>, <título da mensagem>, <botões presentes>, <tipo de mensagem>).
```

A única diferença entre essa sintaxe e a sintaxe da caixa de mensagem (`showMessageDialog()`) se refere ao parâmetro `<botões presentes>` no qual são inseridos os botões que aparecerão na caixa de diálogo.

showInputDialog()

As caixas de diálogo de entrada são geradas a partir do método `showInputDialog()` e são utilizadas para fazer uma pergunta ao usuário e solicitar a entrada de um texto. Esse texto será armazenado em uma variável qualquer do tipo `String`. A sintaxe usada na criação de caixas de diálogo de entrada é praticamente a mesma usada nas caixas de mensagem.

Sintaxe:

```
JOptionPane.showInputDialog(Component, <mensagem>, <título da mensagem>, <botões presentes>, <tipo de mensagem>).
```

showOptionDialog()

As caixas de diálogo de opção são geradas a partir do método `showOptionDialog()` e são bem complexas. São capazes de combinar todos os recursos já vistos nas outras caixas de diálogo.

Sintaxe:

```
JOptionPane.showOptionDialog(Component, <mensagem>, <título da mensagem>, <botões presentes>, <tipo de mensagem>, <ícone>, <array de objetos>, <seleção padrão>).
```

Onde:

- **Botões presentes:** Um tipo de caixa que utiliza as variáveis YES_NO_OPTION, YES_NO_CANCEL_OPTION ou OK_CANCEL_OPTION. Deve-se utilizar o 0(zero) caso outros botões sejam utilizados no lugar destes.
- **Tipo de mensagem:** Uma das variáveis de pergunta, informação, alerta, ou erro;
- **Ícone:** Um objeto ImageIcon a ser apresentado no lugar de um dos ícones do argumento anterior (<tipo de mensagem>). Se nenhum ícone for usado, deve-se colocar a palavra "null" no lugar desse argumento.
- **Array de objetos:** É um array de objetos que contém os componentes ou outros objetos que representam as escolhas da caixa de diálogo, caso YES_NO_OPTION, YES_NO_CANCEL_OPTION ou OK_CANCEL_OPTION não esteja sendo utilizado.
- **Seleção Padrão:** O objeto que representa a seleção padrão, caso as opções YES_NO_OPTION, YES_NO_CANCEL_OPTION ou OK_CANCEL_OPTION não estejam sendo usadas.

Exemplos da utilização dos métodos vistos:

```
import javax.swing.JOptionPane;

public class TesteJOptionPane {
    public static void main(String args[]){
        int resposta = JOptionPane.NO_OPTION;

        while (resposta == JOptionPane.NO_OPTION){
            //Exemplo de utilização do metodo showInputDialog()
            String sexo = JOptionPane.showInputDialog(null, "Digite seu sexo.
            \nM para masculino e F para feminino" , "Definir sexo",
            JOptionPane.PLAIN_MESSAGE);

            if (sexo.equalsIgnoreCase("M")){
                //Exemplo de utilização do metodo showMessageDialog()
                JOptionPane.showMessageDialog(null, "Você é do sexo masculino!",
                "Sexo Masculino", JOptionPane.INFORMATION_MESSAGE);
            }
            else if (sexo.equalsIgnoreCase("F")){
                JOptionPane.showMessageDialog(null, "Você é do sexo feminino!",
                "Sexo Feminino", JOptionPane.INFORMATION_MESSAGE);
            }
            else{
                JOptionPane.showMessageDialog(null, "Sexo inválido", "Invalido",
                JOptionPane.ERROR_MESSAGE);
            }
        }
    }
}
```

```

        //Exemplo de utilizacao do metodo showConfirmDialog()
        resposta = JOptionPane.showConfirmDialog(null, "Deseja encerrar o
programa?", "Encerrar", JOptionPane.YES_NO_OPTION,
JOptionPane.QUESTION_MESSAGE);
    }

    JOptionPane.showMessageDialog(null, "O programa será encerrado",
"Finalizar", JOptionPane.WARNING_MESSAGE);

    System.exit(0);
}
}

```

```

import javax.swing.JOptionPane;

public class TesteJOptionPane2 {
    public static void main(String[] args) {
        String[] escolha = {"entre 1 e 20 anos",
                            "entre 21 e 40 anos",
                            "mais de 40 anos"};

        //Exemplo da utilização do metodo showOptionDialog()
        int resp = JOptionPane.showOptionDialog(null, "Qual sua
idade?", "Idade", 0, JOptionPane.QUESTION_MESSAGE, null, escolha,
escolha[0]);
        String idade = escolha[resp];
        JOptionPane.showMessageDialog(null, "Foi informado: " +
idade, "Informação", JOptionPane.INFORMATION_MESSAGE);
        System.exit(0);
    }
}

```

As Classes I/O

Existe ainda uma outra forma de realizar entrada e saída. As classes I/O `DataInputStream` e `DataOutputStream` podem ser utilizadas em conjunto com os atributos `System.in` e `System.out` para entrada e saída de dados pelo teclado via console.

Exemplo:

```

import java.io.DataOutputStream;
import java.io.DataInputStream;
import java.io.IOException;
public class TesteDataOutputStream {

    public static void main(String[] args) {
        // Utilizacao da classe DataOutputStream em conjunto com System.out
        DataOutputStream saida = new DataOutputStream (System.out);
        // Utilizacao da classe DataInputStream em conjunto com System.in
        DataInputStream entrada = new DataInputStream (System.in);
    }
}

```

```
try{
    byte b[] = new byte[50];
    saida.writeBytes("Digite o seu nome\n");
    entrada.read(b);
    String m = new String(b).trim();
    saida.writeBytes("O nome digitado foi: ");
    saida.writeBytes(m);
    //descarrega (na tela) o que esta armazenado no objeto saida
    saida.flush();
}
catch (IOException e){
    System.out.println();
}
}
```