

# Algoritmos

Prof. Ricardo Luiz de Freitas

[ricardo.freitas@academico.domhelder.edu.br](mailto:ricardo.freitas@academico.domhelder.edu.br)

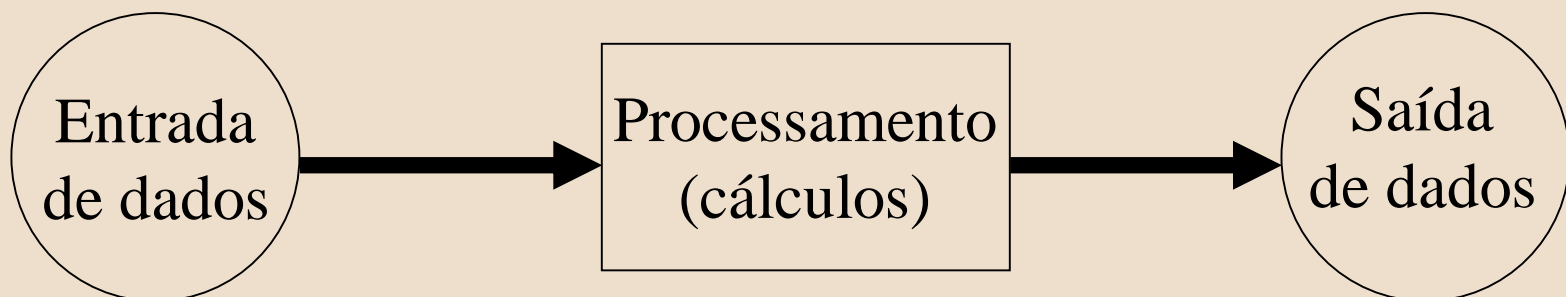


# Conteúdo 4

- Comando de entrada (*input*) de dados
  - Classe **Scanner**
- Comando de saída (*output*) de dados
  - Método **System.out.print**
- Diálogos com a classe **JOptionPane**
  - Métodos: showMessageDialog, showInputDialog e showOptionDialog
- Entrada de dados via **arquivo texto**
  - Classe **Scanner**
- Exercícios

# Etapas de um programa

- 1ª etapa : entrada de dados (informados pelo usuário)
- 2ª etapa : processamento (cálculos)
- 3ª etapa : saída de dados (exibição dos resultados)



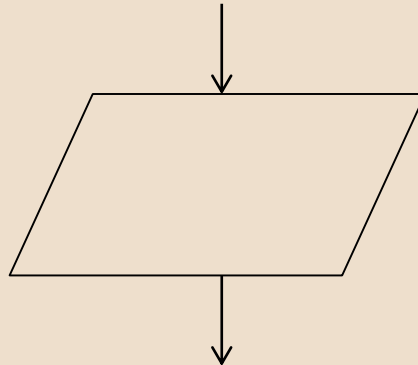
# Entrada de Dados

(comandos de I/O)

- Comandos utilizados para receber valores informados pelo usuário via teclado ou qualquer dispositivo de entrada de dados (teclado, arquivo, etc.), atribuindo-os às variáveis especificadas nos comandos.
- O tipo de dado recebido depende do tipo da variável utilizada no comando, se a variável for do tipo **int** não se pode receber um texto, pois abortará a execução do programa.
- No caso do teclado, o usuário digita o dado e aperta ENTER. O dado é gravado na variável e o cursor da tela pula para a primeira coluna da próxima linha.

# Fluxograma

- Representação da entrada de dados:



# Classe Scanner

(entrada de dados)

- Uma das formas de entrada de dados utilizada no Java é por meio da classe **Scanner** que requer a importação do pacote **java.util.Scanner**:

```
import java.util.Scanner;

public class PrimeiroPrograma {

    public static void main(String[] args) {

        long idade;
        String nome;

        Scanner teclado = new Scanner(System.in);
        System.out.print("Informe seu nome: ");
        nome = teclado.nextLine();
        System.out.print("Informe sua idade: ");
        idade = teclado.nextLong();
        teclado.close();

    }

}
```

# Métodos da classe **Scanner**

(entrada de dados)

Método	Descrição
next()	Aguarda a digitação de um valor do tipo <b>String</b> com <u>uma palavra</u> (sem espaços)
nextLine()	Aguarda a digitação de valor do tipo <b>String</b> , com <u>uma ou mais palavras</u>
next().charAt(0)	Aguarda a digitação de valor do tipo <b>char</b> com <u>apenas um caracter</u>
nextInt()	Aguarda a digitação de um valor do tipo <b>int</b>
nextLong()	Aguarda a digitação de um valor do tipo <b>long</b>
nextByte()	Aguarda a digitação de um valor tipo <b>byte</b>
nextFloat()	Aguarda a digitação de um valor tipo <b>float</b>
nextDouble()	Aguarda a digitação de um valor tipo <b>double</b>
nextBoolean()	Aguarda a digitação de um valor tipo <b>boolean</b> ( <u>true</u> ou <u>false</u> )

# Bug (característica) do Java

(entrada de dados)

- Bug no Java (em versões antigas) quando é feita uma leitura de **String** logo depois da leitura de uma variável numérica (**int**, **long**, **double**, etc.):

```
import java.util.Scanner;

public class SegundoPrograma {

    public static void main(String[] args) {
        long idade;
        String nome;
        Scanner teclado = new Scanner(System.in);
        System.out.print("Informe sua idade: ");
        idade = teclado.nextLong();
        System.out.print("Informe seu nome: ");
        nome = teclado.nextLine();
        teclado.close();
    }
```

**PROBLEMA**



# Bug (característica) do Java

(entrada de dados)

- Bug no Java (em versões antigas) quando é feita uma leitura de **String** logo depois da leitura de uma variável numérica (**int**, **long**, **double**, etc.):

```
import java.util.Scanner;

public class SegundoPrograma {
    public static void main(String[] args) {
        long idade;
        String nome;
        Scanner teclado = new Scanner(System.in);
        System.out.print("Informe sua idade: ");
        idade = teclado.nextLong();
        teclado.nextLine();
        System.out.print("Informe seu nome: ");
        nome = teclado.nextLine();
        teclado.close();
    }
}
```

**Acrescentar  
esta linha!**

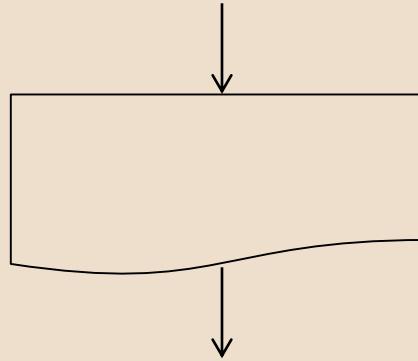
# Saída de Dados

(comandos de I/O)

- Comandos utilizados para se escrever (imprimir) informações na tela ou em qualquer dispositivo de saída de dados (impressora, fax, modem, etc.).

# Fluxograma

- Representação da saída de dados:



# Objeto **System.out**

(saída de dados)

- Uma das formas de saída de dados utilizada no Java é por meio dos métodos presentes no objeto **System.out**.
- **System.out.print(...);**
  - A informação entre parênteses é impressa e o cursor da tela permanece na mesma linha.
- **System.out.println(...);**
  - Idem, porém o cursor da tela pula para a primeira coluna da próxima linha.
- **System.out.printf(...);**
  - A informação entre parênteses é impressa conforme uma formatação.

# Objeto **System.out**

(saída de dados)

- Temos as seguintes variações para os comandos (métodos):
  - **System.out.println(VARIÁVEL);**
    - O conteúdo na memória da VARIÁVEL é impresso na tela.
  - **System.out.println("Mensagem a ser impressa");**
    - O texto entre aspas é impresso da forma como foi digitado.
  - **System.out.println("Mensagem 1 "+VAR1+" Mensagem 2 "+VAR2);**
    - Os textos são impressos junto com os conteúdos das variáveis VAR1 e VAR2, na mesma linha e um depois do outro totalmente colados uns aos outros.
  - **System.out.println();**
    - Salto de uma linha em branco.
- **ATENÇÃO:** É importante o uso de espaços dentro das aspas a fim de que os dados (textos e conteúdos de variáveis) sejam impressos separados uns dos outros.

# Sequências de escape do **print**

(saída de dados)

- Quando aparece uma barra invertida “\” em uma string de caracteres, o Java combina o próximo caractere com as barras invertidas para formar uma **sequência de escape**.

Sequências de Escape		
<b>\n</b>	Nova linha	Posiciona o cursor da tela no início da próxima linha
<b>\t</b>	Tabulação horizontal	Move o cursor de tela para a próxima parada de tabulação
<b>\r</b>	Retorno de carro	Posiciona o cursor da tela no início da linha atual (não avança para a próxima linha).
<b>\\</b>	Barras invertidas	Utilizada para imprimir um caractere de barra invertida.
<b>\"</b>	Aspas duplas	Utilizada para imprimir um caractere de aspas duplas.
<b>%%</b>	Porcentagem	Utilizado para imprimir um caractere de porcentagem.

# Sequências de escape do **print**

(saída de dados)

## ■ Exemplo de **sequências de escape**:

```
System.out.println("Bem vindo \nao curso \nda Dom Helder!");
```

**Impressão:** Bem vindo  
ao curso  
da Dom Helder!

Os vários `\n` no meio do texto que está entre as aspas fará com que a impressão do texto pule para a primeira coluna da próxima linha de impressão.

# Especificador de formato do **printf**

(saída de dados)

- **Especificador de formato** é um marcador de lugar para um valor e especifica o tipo da saída de dados;
- Eles se iniciam com um sinal de porcentagem (%) e são seguidos por um caractere que representa o tipo de dados.

Especificadores de Formato	
<b>%tamanho</b>	Decimal inteiro
<b>%tamanho</b> <b>s</b>	String
<b>%c</b>	Caractere
<b>%b</b>	Booleano
<b>%tamanho.casasDecimaisf</b>	Decimal com casas decimais

- **tamanho** é a quantidade de colunas que o ocupará a impressão do dado na tela (incluindo o ponto e as casas decimais);
- **casasDecimais** é a quantidade de casas decimais que serão impressas na tela. Se o número de casas do dado for maior, arredonda, se for menor completa com zeros à direita.



# Especificador de formato do **printf**

(saída de dados)

- Exemplo 1 de **especificador de formato**:

**double** valor = 123.569;

**System.out.printf**("Valor real: %7.2f", valor);

**Impressão:** Valor real: 123,57

**System.out.printf**("Valor real: %11.5f", valor);

**Impressão:** Valor real: 123,56900

**System.out.printf**("Valor real: %5.0f", valor);

**Impressão:** Valor real: 124

O formato **%11.5f** está ligado a variável **double** **valor** e seu conteúdo, que será impresso na tela ocupando 11 colunas (incluindo a parte inteira do conteúdo, a vírgula decimal e a parte decimal com 5 casas arredondadas, se for o caso) completando com brancos à esquerda e zeros à direita.

- O primeiro valor depois do argumento % especifica o tamanho que o campo ocupará na tela e o segundo o número de casas decimais, arredondando o valor ou acrescentando zeros à direita, dependendo do caso;

# Especificador de formato do **printf**

(saída de dados)

## ■ Exemplo 2 de **especificador de formato**:

```
int seq = 37;
```

```
String nome = "Joselito Sem Noção";
```

```
System.out.printf("Nome do Fulano: %d\n%s\n",seq,nome);
```



Impressão: Nome do Fulano: 37  
Joselito Sem Noção

- A chamada do método do exemplo acima especifica dois argumentos separados por vírgulas (variáveis **seq** e **nome**):
  - ❑ O primeiro argumento (**seq**), que é uma variável numérica inteira, substituirá o **%d**;
  - ❑ O segundo argumento (**nome**), que é uma variável string, substituirá o **%s**.

# Especificador de formato do **printf**

(saída de dados)

- Exemplo 3 de **especificador de formato**:

**int** i = 10;

**String** s = "formatação";

**double** d = 123.456;

**char** c = 'a';

**boolean** b = true;

**System.out.printf**("Dados: %d - %s - %7.2f - %c - %b \n", i, s, d, c, b);

**Impressão:** Dados: 10 - formatação - 123,47 - a – true

O formato **%7.2f** está ligado a variável **double d** e seu conteúdo, que será impresso na tela ocupando 7 colunas (incluindo a parte inteira do conteúdo, a vírgula decimal e a parte decimal com 2 casas arredondadas, se for o caso) completando com brancos à esquerda e zeros à direita.

# Formatando com DecimalFormat

- O Java oferece várias maneiras de formatar números. Uma delas é através da classe **DecimalFormat**:

```
import java.text.DecimalFormat;  
public class FormatandoNumeros1 {  
    public static void main(String[] args) {  
        double valor = 12005.867;  
        String valorFormatado;  
        valorFormatado = new DecimalFormat("#,##0.00").format(valor);  
        System.out.println(valorFormatado);  
        // Imprimirá: 12.005,87  
        valorFormatado = new DecimalFormat("R$ #,##0.00").format(valor);  
        System.out.println(valorFormatado);  
        // Imprimirá: R$ 12.005,87  
        valorFormatado = new DecimalFormat("#,##0").format(valor);  
        System.out.println(valorFormatado);  
        // Imprimirá: 12.006  
    }  
}
```

Arredondará para duas casas decimais.

# Formatando com NumberFormat

■ Outra forma de formatar números é através da classe **NumberFormat**:

```
import java.text.NumberFormat;  
public class FormatandoNumeros2 {  
    public static void main(String[] args) {  
        double valor = 12005.867;  
        String valorFormatado;  
        valorFormatado = NumberFormat.getNumberInstance().format(valor);  
        System.out.println(valorFormatado);  
        // Imprimirá: 12.005,867  
        valorFormatado = NumberFormat.getIntegerInstance().format(valor);  
        System.out.println(valorFormatado);  
        // Imprimirá: 12.006  
        valorFormatado = NumberFormat.getCurrencyInstance().format(valor);  
        System.out.println(valorFormatado);  
        // Imprimirá: R$ 12.005,87  
    }  
}
```

Pegará a configuração corrente do Windows de formatação de moeda

# Formatando com NumberFormat

- Formatar números percentuais com a classe **NumberFormat**:

```
import java.text.NumberFormat;  
public class FormatandoNumeros2 {  
    public static void main(String[] args) {  
        double valor = 0.5;  
        String valorFormatado;  
        valorFormatado = NumberFormat.getPercentInstance().format(valor);  
        System.out.println(valorFormatado);  
        // Imprimirá: 50%  
    }  
}
```

# Formatando com `String.format`

- Também podemos utilizar o método **format** da classe **String** para formar números (funciona igual ao **printf**, ou seja, não coloca separador de milhar):

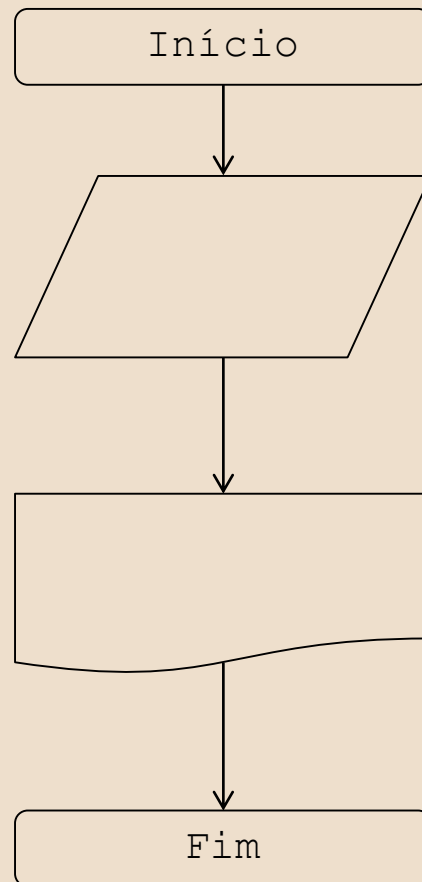
```
public class FormatandoNumeros3 {  
    public static void main(String[] args) {  
        double valor = 12005.867;  
        String valorFormatado;  
        valorFormatado = String.format("%1.2f", valor);  
        System.out.println(valorFormatado);  
        // Imprimirá: 12005,87  
        valorFormatado = String.format("%1.0f", valor);  
        System.out.println(valorFormatado);  
        // Imprimirá: 12006  
    }  
}
```

Arredondará para duas casas decimais.

Arredondará para zero casas decimais.

# Comandos de IO

## Fluxograma





# Exemplo 1

(Programa resolvido)

```
import java.util.Scanner;

public class CalcSomaMedia2 {

    public static void main(String[] args) {
        int n1, n2, soma;
        double media;

        Scanner teclado = new Scanner(System.in);
        System.out.print("Digite um número: ");
        n1 = teclado.nextInt();
        System.out.print("Digite outro número: ");
        n2 = teclado.nextInt();

        soma = n1 + n2;
        media = (float) soma / 2;

        System.out.println("Soma: "+soma);
        System.out.println("Média: "+media);
        teclado.close();
    }
}
```

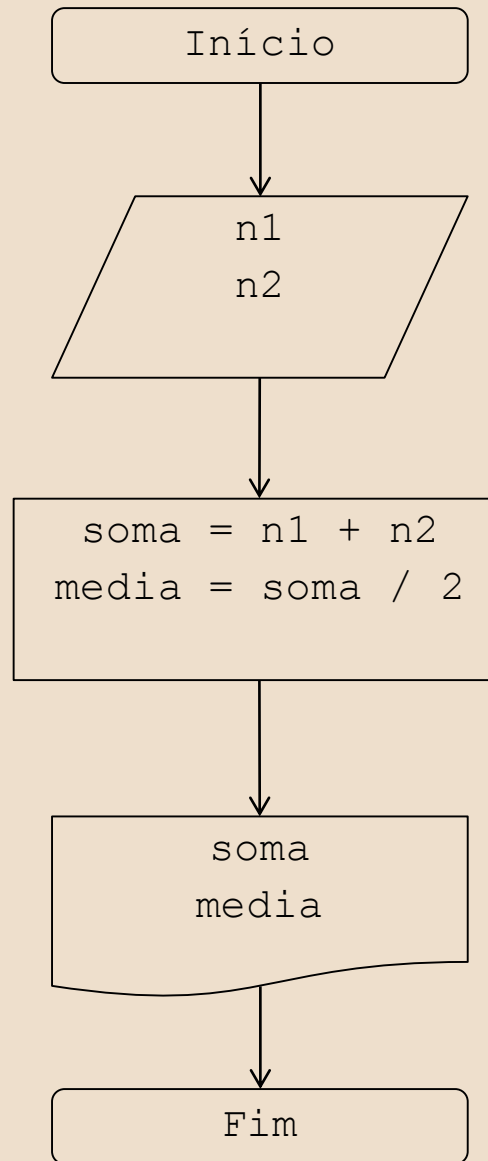
Entrada  
de Dados

Cálculos

Saída de  
Dados

# Exemplo 1

## Fluxograma



## Exemplo 2

Faça um programa que solicite ao usuário o seu *nome completo* e o *número de filhos* que ele tem, e depois imprima estas informações na tela do computador.

# Exemplo 2

(Programa resolvido)

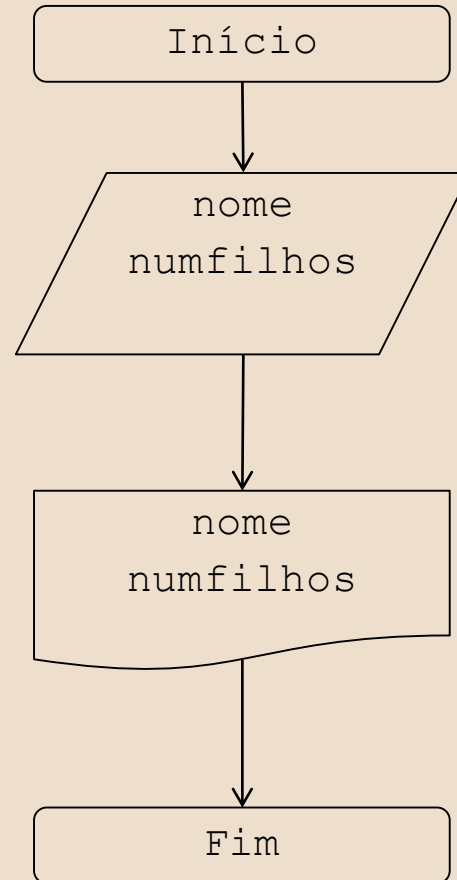
```
import java.util.Scanner;
public class NomeFilhos {
    public static void main(String[] args) {
        String nome;
        int numFilhos;
        Scanner teclado = new Scanner(System.in);
        System.out.print("Qual seu nome? ");
        nome = teclado.nextLine();
        System.out.print("Quantos filhos você tem? ");
        numFilhos = teclado.nextInt();
        System.out.println("Nome: "+nome);
        System.out.println("Quantidade de filhos: "+numFilhos);
        teclado.close();
    }
}
```

Entrada  
de Dados

Saída de  
Dados

# Exemplo 2

## Fluxograma



# Exemplo 3

Faça um programa que solicite ao usuário o seu *nome*, seu endereço completo (*logradouro*, *número*, *bairro*, *cidade*, *estado* e *CEP*), seu *peso* (em Kg) e o *número do sapato*. Depois imprima uma etiqueta com estas informações de acordo com o layout abaixo:

```
x-----nome-----x
x-----logradouro-----x, x---numero---x
Bairro x-----bairro-----x
x-----cidade-----x – x--estado—x
CEP x---cep---x
```

Peso (kg): x---peso---x

Número do sapato: x--numero do sapato--x

# Exemplo 3

Exemplo de saída de dados:

RICARDO LUIZ DE FREITAS  
RUA ALVARES MACIEL, 628  
**Bairro** SANTA EFIGÊNIA  
BELO HORIZONTE – MG  
**CEP** 30150-250

**Peso (kg):** 86.5

**Número do sapato:** 40

# Exemplo 3

(Programa resolvido)

```
import java.util.Scanner;

public class ImprimeEtiqueta {
    public static void main(String[] args) {
        String nome, logradouro, bairro, cidade, uf, cep, numero;
        int sapato;
        float peso;
        Scanner teclado = new Scanner(System.in);
        System.out.print("Informe seu nome: ");
        nome = teclado.nextLine();
        System.out.print("Informe seu logradouro: ");
        logradouro = teclado.nextLine();
        System.out.print("Informe seu número: ");
        numero = teclado.nextLine();
        System.out.print("Informe seu bairro: ");
        bairro = teclado.nextLine();
        System.out.print("Informe sua cidade: ");
        cidade = teclado.nextLine();
    }
}
```



# Exemplo 3

(Programa resolvido)

```
System.out.print("Informe a UF: ");
uf = teclado.nextLine();
System.out.print("Informe seu CEP: ");
cep = teclado.nextLine();
System.out.print("Informe número do seu sapato: ");
sapato = teclado.nextInt();
System.out.print("Informe seu peso: ");
peso = teclado.nextFloat();
System.out.println();
System.out.println(nome);
System.out.println(logradouro+", "+numero);
System.out.println("Bairro "+bairro);
System.out.println(cidade+" - "+uf);
System.out.println("CEP "+cep);
System.out.println();
System.out.println("Peso (Kg): "+sapato);
System.out.println("Número do sapato: "+peso);
teclado.close();
```

# Exercício 1

Faça um programa que solicite ao usuário o seu *primeiro nome*, seu *nome do meio*, seu *sobrenome*, sua *idade* e depois imprima todos estes dados de acordo com o layout abaixo:

Sobrenome, PrimeiroNome SegundoNome

Idade: 99 anos.

Exemplo de saída de dados (console):

Freitas, Ricardo Luiz

Idade: 50 anos.

## Exercício 2

Faça um programa que solicite ao usuário o seu *nome*, seus documentos (*CPF*, *identidade*, *título de eleitor*, *carteira de motorista*), seu *salário* e o *nome da empresa* em que trabalha, e depois imprima a sua FICHA FUNCIONAL de acordo com o layout abaixo:

FICHA FUNCIONAL DE: x-----nome-----x

Documentos:

CPF .....	x-----cpf-----x
C.I. ....	x---identidade---x
Título de eleitor .....	x---título de eleitor---x
Carteira de motorista .....	x---carteira de motorista ---x

Empresa: x-----nome da empresa-----x

Salário: R\$ x-----salário-----x

## Exercício 2

Exemplo de saída de dados (console:

FICHA FUNCIONAL DE: RICARDO LUIZ DE FREITAS

Documentos:

CPF ..... 415467889-03

C.I. .... M5677823

Título de eleitor ..... 458690688444

Carteira de motorista ..... 22349377

Empresa: EMGE

Salário: R\$32500.45

# Exercício 3

A Secretaria do Meio Ambiente mediu a quantidade de poluentes atmosféricos emitidos por uma empresa. Dependendo do valor obtido, a empresa foi multada conforme a tabela abaixo. Faça um programa que monte esta tabela, solicitando as informações variáveis (xxx) para o usuário:

---

## Quantidade de Poluente Emitido x Valor da Multa

---

Até **xxx** <sup>(1)</sup> multa de R\$ **xxx** <sup>(3)</sup>

---

Entre **xxx** <sup>(1)</sup> e **xxx** <sup>(2)</sup> multa de R\$ **xxx** <sup>(4)</sup>

---

Acima de **xxx** <sup>(2)</sup> multa de R\$ **xxx** <sup>(5)</sup> por poluente emitido

---

# Exercício 3

Exemplo de saída de dados (console):

---

## Quantidade de Poluente Emitido x Valor da Multa

---

Até **1500** multa de R\$**1000,55**

---

Acima de **1500** até **3000** multa de R\$**3550,34**

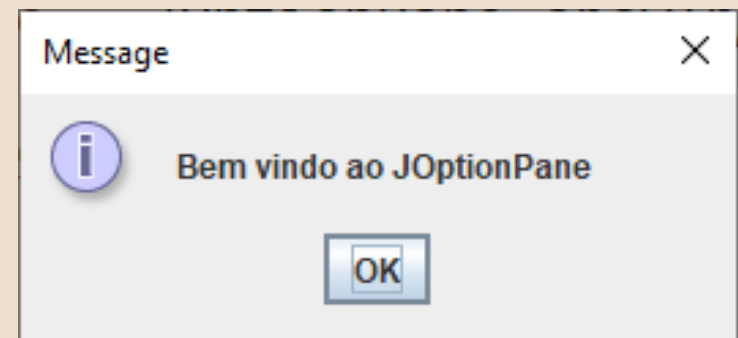
---

Acima de **3000** multa de R\$**1200,21** por poluente emitido

---

# Diálogos com a classe JOptionPane

- Até agora vimos a classe **Scanner** para entrada de dados, e o método **System.out.println** para escrever informações na tela, ambos via console;
- A linguagem Java oferece diversas formas de interação com o usuário, a grande maioria em janelas;
- Para se evitar a criação de uma interface completa, pode-se utilizar as chamadas caixas de diálogo.



# Diálogos com a classe JOptionPane

- **JOptionPane** é uma classe da biblioteca de interfaces gráficas do **javax.swing**;
- Para a utilizar dentro de seu programa (ou classe), a classe abaixo deve ser importada:

- ❑ `import javax.swing.JOptionPane;`

- **Métodos:**

- ❑ `showMessageDialog;`
  - ❑ `showInputDialog;`
  - ❑ `showConfirmDialog;`
  - ❑ `showOptionDialog.`







# Método showMessageDialog

- Exibir uma caixa de mensagem para mostrar ao usuário alguma mensagem.

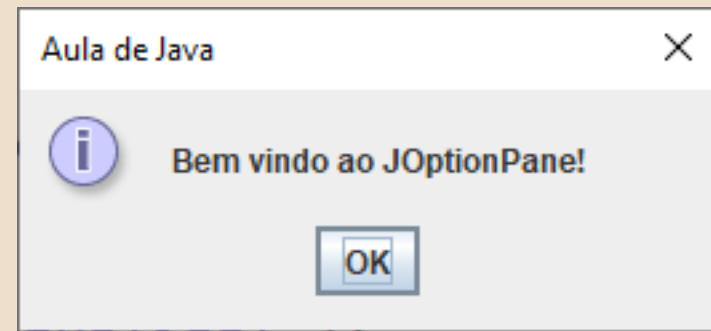
```
JOptionPane.showMessageDialog(null,
    mensagem,           // mensagem do corpo da caixa
    título_da_caixa,    // título da caixa
    tipo_do_diálogo);   // ver abaixo
```

□ Tipos de diálogo (ícone da caixa):

- |                       |   |                          |
|-----------------------|---|--------------------------|
| ■ ERROR_MESSAGE       |   | // Ícone de erro         |
| ■ INFORMATION_MESSAGE |  | // Ícone de informação   |
| ■ WARNING_MESSAGE     |  | // Ícone de aviso        |
| ■ QUESTION_MESSAGE    |  | // Ícone de interrogação |
| ■ PLAIN_MESSAGE       |   | // Nenhum ícone          |

# Método showMessageDialog

```
import javax.swing.JOptionPane;  
public class TelaMensagem1 {  
    public static void main(String[] args) {  
        JOptionPane.showMessageDialog(null,  
            "Bem vindo ao JOptionPane!",  
            "Aula de Java",  
            JOptionPane.INFORMATION_MESSAGE);  
    }  
}
```







# Método `showInputDialog` (1)

- Exibir uma caixa de entrada de dados, retornando o texto (**string**) digitado pelo usuário.

```
String texto = JOptionPane.showInputDialog(null,  
    mensagem,                // mensagem do corpo da caixa  
    título_da_caixa,         // título da caixa  
    tipo_do_diálogo);        // ver abaixo
```

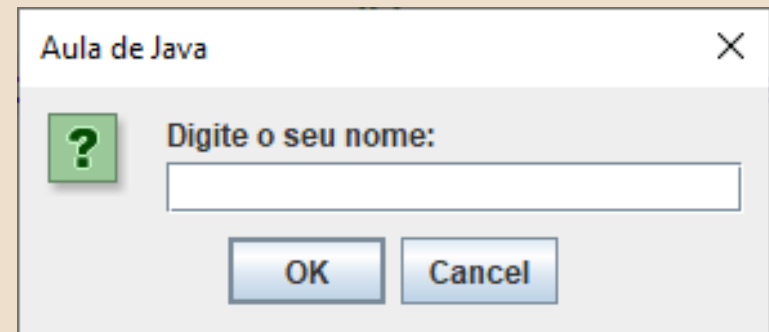
## ☐ Tipos de diálogos:

- |                                    |   |                          |
|------------------------------------|---|--------------------------|
| ■ <code>ERROR_MESSAGE</code>       |   | // Ícone de erro         |
| ■ <code>INFORMATION_MESSAGE</code> |  | // Ícone de informação   |
| ■ <code>WARNING_MESSAGE</code>     |  | // Ícone de aviso        |
| ■ <code>QUESTION_MESSAGE</code>    |  | // Ícone de interrogação |
| ■ <code>PLAIN_MESSAGE</code>       |   | // Nenhum ícone          |

# Método `showInputDialog` (1)

```
import javax.swing.JOptionPane;
public class TelaMensagem2 {
    public static void main(String[] args) {
        String nome;
        nome = JOptionPane.showInputDialog(null,
            "Digite seu nome:",
            "Aula de Java",
            JOptionPane.QUESTION_MESSAGE);
        System.out.println(nome);
    }
}
```

A variável **nome** irá receber o retorno do método, ou seja, o que o usuário digitou, sempre como String.







# Método `showInputDialog` (2)

- Exibir uma caixa de entrada de dados, retornando o texto (**string**) selecionado pelo usuário.

```
String texto = JOptionPane.showInputDialog(null,  
    mensagem,                // mensagem do corpo da caixa  
    título_da_caixa,         // título da caixa  
    tipo_do_diálogo,  
    null,  
    label_das_opções,  
    label_da_opção_default);
```

# Método showInputDialog (2)

## ❑ Tipos de diálogos:

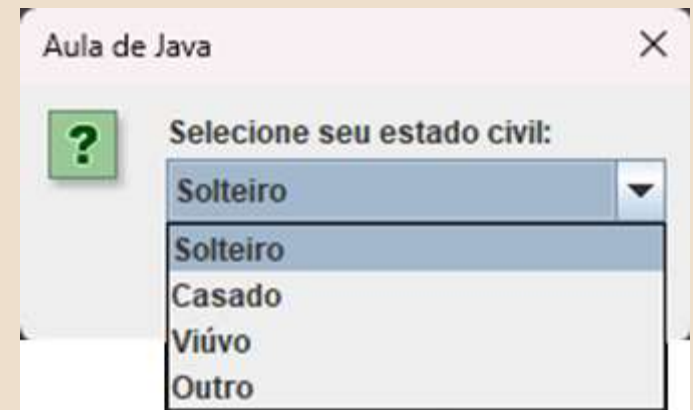
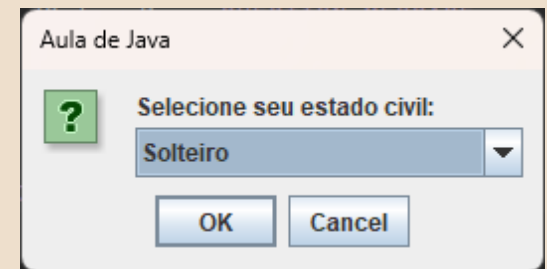
- `ERROR_MESSAGE`  `// Ícone de erro`
- `INFORMATION_MESSAGE`  `// Ícone de informação`
- `WARNING_MESSAGE`  `// Ícone de aviso`
- `QUESTION_MESSAGE`  `// Ícone de interrogação`
- `PLAIN_MESSAGE` `// Nenhum ícone`

```
String[] label_das_opcoes = {"opção 1", "opção 2", "opção 3", ...};  
String label_da_opcao_default = "opção 1";
```

# Método showInputDialog (2)

```
import javax.swing.JOptionPane;
public class TelaMensagem5 {
    public static void main(String[] args) {
        String[] opcoes = {"Solteiro", "Casado", "Viúvo", "Outro"};
        Object nome;
        nome = JOptionPane.showInputDialog(null,
            "Selecione seu estado civil:",
            "Aula de Java",
            JOptionPane.QUESTION_MESSAGE,
            null,
            opcoes,
            "Solteiro");
        System.out.println(nome);
    }
}
```

A variável **nome** irá receber o retorno do método, ou seja, o que o usuário selecionou, sempre como String



# Método `showOptionDialog`





- Exibir uma caixa de seleção, retornando o índice (0..?) do botão selecionado pelo usuário.

```
int indice = JOptionPane.showOptionDialog(null,  
    mensagem,                // mensagem do corpo da caixa  
    título_da_caixa,         // título da caixa  
    0,  
    tipo_do_diálogo,        // ver próximo slide  
    null,  
    label_dos_botões,        // Labels dos botões  
    label_do_botão_default); // Label do botão default
```



# Método showDialog

## ❑ Tipos de diálogos:

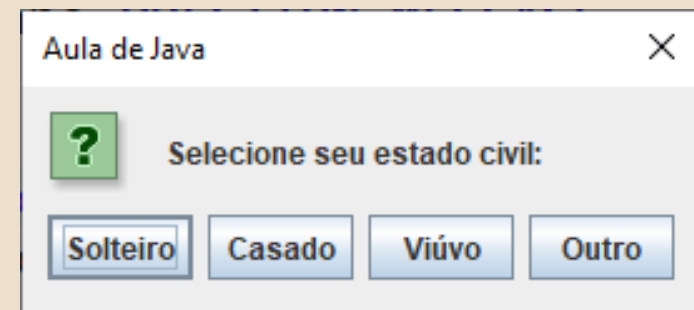
■ ERROR_MESSAGE		// Ícone de erro
■ INFORMATION_MESSAGE		// Ícone de informação
■ WARNING_MESSAGE		// Ícone de aviso
■ QUESTION_MESSAGE		// Ícone de interrogação
■ PLAIN_MESSAGE		// Nenhum ícone

```
String[] label_dos_botoes = {"botão 1", "botão 2", "botão 3", ...};  
String label_do_botao_default = "botão 1";
```

# Método showDialog

```
import javax.swing.JOptionPane;
public class TelaMensagem3 {
    public static void main(String[] args) {
        String[] botoes = {"Solteiro", "Casado", "Viúvo", "Outro"};
        int indice;
        indice = JOptionPane.showMessageDialog(null,
            "Selecione seu estado civil:",
            "Aula de Java",
            0,
            JOptionPane.QUESTION_MESSAGE,
            null,
            botoes,
            "Solteiro");
        System.out.println(botoes[indice]);
    }
}
```

A variável **indice** irá receber o retorno do método, ou seja, o índice do botão que o usuário clicou, sempre como int.



# Método `showConfirmDialog`

- Exibir uma caixa de seleção, retornando o índice (**0..?**) do botão selecionado pelo usuário.

```
int indice = JOptionPane.showConfirmDialog(null,  
    mensagem,                // mensagem do corpo da caixa  
    título_da_caixa,         // título da caixa  
    tipo_de_botão);          // ver abaixo
```

## ☐ Tipos de botões:

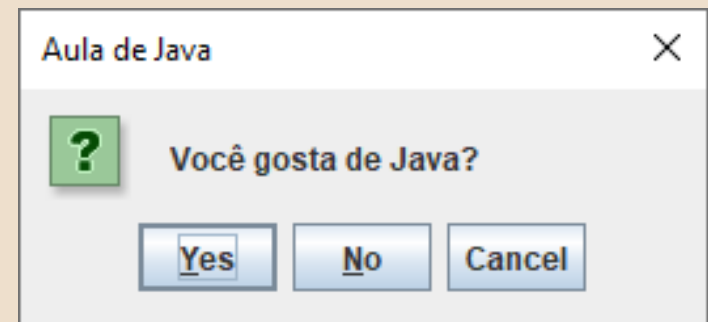
- `YES_NO_OPTION` // YES - NO
- `YES_NO_CANCEL_OPTION` // YES - NO - CANCEL
- `OK_CANCEL_OPTION` // OK - CANCEL

# Método `showConfirmDialog`

- Exibir uma caixa de mensagem para mostrar ao usuário alguma mensagem.

```
import javax.swing.JOptionPane;

public class TelaMensagem4 {
    public static void main(String[] args) {
        int indice;
        indice = JOptionPane.showConfirmDialog(null,
            "Você gosta de Java?",
            "Aula de Java",
            JOptionPane.YES_NO_CANCEL_OPTION);
        System.out.println(indice);
    }
}
```



## Exemplo 4

Faça um programa que solicite ao usuário o seu *nome completo* e o *número de filhos* que ele tem, e depois imprima estas informações na tela do computador.

# Exemplo 2 (anterior – versão 1)

(Programa resolvido)

```
import java.util.Scanner;

public class NomeFilhos {
    public static void main(String[] args) {
        String nome;
        int numFilhos;

        Scanner teclado = new Scanner(System.in);
        System.out.print("Qual seu nome? ");
        nome = teclado.nextLine();
        System.out.print("Quantos filhos você tem? ");
        numFilhos = teclado.nextInt();
        System.out.println("Nome: "+nome);
        System.out.println("Quantidade de filhos: "+numFilhos);
        teclado.close();
    }
}
```

# Exemplo 2 (anterior – versão 2)

(Programa resolvido)

```
import java.util.Scanner;
public class NomeFilhos {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        System.out.print("Qual seu nome? ");
        String nome = teclado.nextLine();
        System.out.print("Quantos filhos você tem? ");
        int numFilhos = teclado.nextInt();
        System.out.println("Nome: "+nome);
        System.out.println("Quantidade de filhos: "+numFilhos);
        teclado.close();
    }
}
```

# Exemplo 4

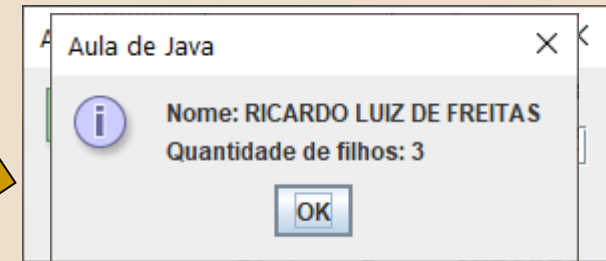
(Programa resolvido)

```
import javax.swing.JOptionPane;

public class NomeFilhos2 {

    public static void main(String[] args) {
        String nome = JOptionPane.showInputDialog(null,
            "Qual seu nome?",
            "Aula de Java",
            JOptionPane.QUESTION_MESSAGE);
        String numFilhosStr = JOptionPane.showInputDialog(null,
            "Quantos filhos você tem?",
            "Aula de Java",
            JOptionPane.QUESTION_MESSAGE);
        int numFilhos = Integer.valueOf(numFilhosStr);
        JOptionPane.showMessageDialog(null,
            "Nome: "+nome+"\nQuantidade de filhos: "+numFilhos,
            "Aula de Java",
            JOptionPane.INFORMATION_MESSAGE);
    }
}
```

A sequência de escape `\n` faz a mensagem ser quebrada numa nova linha do corpo da caixa de diálogo.





# Entrada de dados via Arquivo Texto

- O Java oferece várias maneiras de ler arquivos. Uma delas é através do método **nextLine()** da classe **Scanner**:

```
import java.util.Scanner;

public class LerArqTxt1 {
    public static void main(String[] args) {
        String linha;
        Scanner arquivo = new Scanner(
            LerArqTxt1.class.getResourceAsStream("nomes.txt"));
        linha = arquivo.nextLine();
        System.out.println("Linha 1: "+linha);
        linha = arquivo.nextLine();
        System.out.println("Linha 2: "+linha);
        arquivo.close();
    }
}
```

O arquivo txt tem que estar gravado na pasta onde ficam os arquivos **.class** do projeto.

# Entrada de dados via Arquivo Texto

- Pode-se utilizar uma **estrutura de repetição (while)** para ler todas as linhas (registros) do arquivo, sem que se saiba quantas linhas tem o arquivo:

```
import java.util.Scanner;
public class LerArqTxt2 {
    public static void main(String[] args) {
        String linha;
        Scanner arquivo = new Scanner(
            LerArqTxt2.class.getResourceAsStream("nomes.txt"));
        while (arquivo.hasNextLine()) {
            linha = arquivo.nextLine();
            System.out.println(linha);
        }
        arquivo.close();
    }
}
```

## Exemplo 5

Faça um programa que leia de um arquivo texto (**pessoa.txt**) o *nome completo*, a *quantidade de filhos*, o *salário* e o *nome da mãe* de uma pessoa, e depois imprima estas informações na tela do computador.

# Exemplo 5

(Programa resolvido)

```
import java.util.Scanner;
public class NomeFilhos3 {
    public static void main(String[] args) {
        String nome, nomeMae;
        int numFilhos;
        double salario;
        Scanner arquivo = new Scanner(
            NomeFilhos3.class.getResourceAsStream("pessoa.txt"));
        nome = arquivo.nextLine();
        numFilhos = arquivo.nextInt();
        salario = arquivo.nextDouble();
        nomeMae = arquivo.nextLine();
        System.out.println("Nome: "+nome);
        System.out.println("Quantidade de filhos: "+numFilhos);
        System.out.println("Salário R$"+salario);
        System.out.println("Mãe: "+nomeMae);
        arquivo.close();
    }
}
```

Um comando **arquivo.next** para cada dado a ser lido do arquivo, onde cada dado está numa linha.

**PROBLEMA**

# Exemplo 5

(Programa resolvido)

```
import java.util.Scanner;
public class NomeFilhos3 {
    public static void main(String[] args) {
        String nome, nomeMae;
        int numFilhos;
        double salario;
        Scanner arquivo = new Scanner(
            NomeFilhos3.class.getResourceAsStream("pessoa.txt"));
        nome = arquivo.nextLine();
        numFilhos = arquivo.nextInt();
        salario = arquivo.nextDouble();
        arquivo.nextLine();
        nomeMae = arquivo.nextLine();
        System.out.println("Nome: "+nome);
        System.out.println("Quantidade de filhos: "+numFilhos);
        System.out.println("Salário R$"+salario);
        System.out.println("Mãe: "+nomeMae);
        arquivo.close();
    }
}
```

**Acrescentar  
esta linha!**

## Exercício 4

Faça um programa que solicite ao usuário o seu *primeiro nome*, seu *nome do meio*, seu *sobrenome*, sua *idade* e depois imprima todos estes dados de acordo com o layout abaixo:

Sobrenome, PrimeiroNome SegundoNome

Idade: 99 anos.

Exemplo de saída de dados:

Freitas, Ricardo Luiz

Idade: 50 anos.

UTILIZE CAIXAS DE DIÁLOGO PARA INTERAÇÃO COM O USUÁRIO.

# Exercício 5

Faça um programa que **LEIA DE UM ARQUIVO TEXTO (ficha.txt)** o *nome*, documentos (*CPF*, *identidade*, *título de eleitor*, *carteira de motorista*), *salário* e o *nome da empresa* em que uma pessoa trabalha, e depois imprima a FICHA FUNCIONAL dela de acordo com o layout abaixo:

FICHA FUNCIONAL DE: x-----nome-----x

Documentos:

CPF .....	x-----cpf-----x
C.I. ....	x---identidade---x
Título de eleitor .....	x---título de eleitor---x
Carteira de motorista .....	x---carteira de motorista ---x

Empresa: x-----nome da empresa-----x

Salário: R\$ x-----salário-----x

UTILIZE CAIXAS DE DIÁLOGO PARA A SAÍDA DE DADOS.

# Fim

Prof. Ricardo Luiz de Freitas

[ricardo.freitas@academico.domhelder.edu.br](mailto:ricardo.freitas@academico.domhelder.edu.br)