

# 1 Introdução

Neste material estudaremos os algoritmos de sequência simples, ou seja, aqueles que envolvem um conjunto de entradas, processamento e um conjunto de saídas. Faremos a implementação desses algoritmos utilizando a linguagem Java. Para tanto, veremos também declaração de variáveis e métodos de entrada e saída.

## 2 Desenvolvimento

**2.1 Meu primeiro programa em Java** Nosso primeiro programa em Java será aquele exibido no Bloco de Código 2.1.

Bloco de Código 2.1.1

```
public class HelloWorld{  
    public static void main(String[] args) {  
        System.out.println ("Hello, World");  
    }  
}
```

O código de um programa de computador escrito na linguagem Java é compilado para uma forma intermediária de código denominada bytecode, que é interpretada pelas Máquinas Virtuais Java. Para isso, utilizamos o comando `javac` ou acionamos o item do IDE que o invoca. A sintaxe do comando é

**`javac arquivo.java`**

Pronto, agora o nosso bytecode está pronto para ser interpretado pela JVM. Eles estão armazenados em um arquivo de extensão `.class` que foi gerado pelo compilador. Para colocar o programa em execução, use

**`java arquivo`**

Mais precisamente, esse comando coloca uma instância da máquina virtual Java em execução e ela se encarrega de interpretar os bytecodes, explicando para o computador o que ele deve fazer a cada instrução. Repare que não colocamos extensões quando ela é colocada em execução.

**2.2 O arquivo HelloWorld.java** O exemplo mostra que a classe HelloWorld tem um método main que é quase sempre escrito da forma apresentada. Ele também admite algumas variações que veremos conforme aprofundamos os estudos. O método println exibe uma mensagem em tela de comando. Vamos praticar algumas variações, utilizando a classe JOptionPane, por exemplo.

### **2.3 Observações importantes e boas práticas de programação**

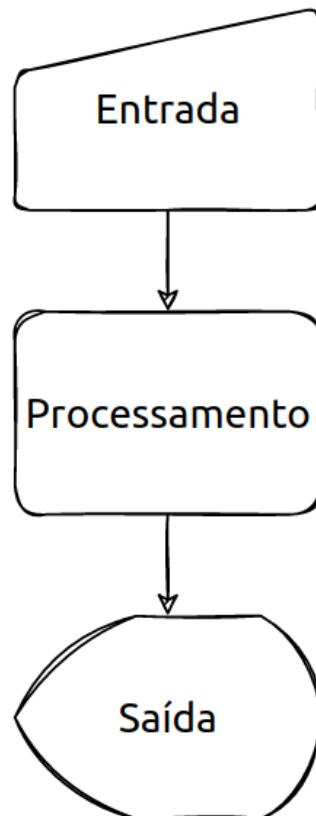
- Blocos são delimitados por {}, a menos que tenham uma única instrução. Neste caso, o uso de {} é opcional.
- Uma boa prática de programação é indentar sempre programas.
- Classes têm nomes iniciados por letra em caixa alta.
- Variáveis e métodos têm nome iniciados por letras em caixa baixa.
- Métodos sempre têm parênteses para parâmetros, mesmo que fiquem vazios.
- Para quaisquer nomes, respeitamos o padrão conhecido como **CamelCase**.

**2.4 Algoritmos Básicos** Um algoritmo básico ou de sequência simples tem a seguinte estrutura:

entrada – processamento – saída

A Figura 2.4.1 ilustra um fluxograma para essa estrutura.

Figura 2.4.1



**2.5 Um exemplo simples** Somar 2 números escolhidos pelo usuário e exibir o resultado.

**Entrada:** informações que são fornecidas ao programa para que ele seja executado. Nesse primeiro exemplo, são os 2 números escolhidos pelo usuário. Nesta fase, o programa irá:

- ler os valores digitados pelo usuário
- armazenar os valores em variáveis denominadas primeiroValor e segundoValor

**Processamento:** Aquilo que o programa faz com a entrada para obter a saída desejada. Nesta fase, o programa irá:

- calcular a soma entre primeiroValor e segundoValor
- armazenar o resultado calculado em uma variável chamada resultado

**Saída:** entrega do resultado obtido. Neste exemplo, o programa irá:

- exibir o valor resultante na saída padrão

**2.6 Armazenamento dos valores na memória: variáveis e tipos primitivos** Uma variável é uma área reservada na memória RAM, identificada por um nome, que pode armazenar valores de um determinado tipo. Um tipo de dado define um conjunto de valores e um conjunto de operações válido. Pelo fato de especificarmos os tipos de nossas variáveis ainda em **tempo de compilação**, ou seja, entregarmos essa informação ao compilador, dizemos que a linguagem Java é **estaticamente tipada**.

No Java temos vários tipos chamados primitivos. Veja alguns bastante comuns:

- **int:** é o tipo de dado capaz de armazenar 32 bits, ou seja, de representar um número inteiro qualquer entre -2.147.483.648 e 2.147.483.647.
- **double:** permite armazenar valores de ponto flutuante IEEE 754 de 64 bits e dupla precisão. Essa é a opção padrão para valores decimais.
- **boolean:** armazena um único byte de informação, que pode ser representado pelas palavras false (falso) ou true (verdadeiro).

O Bloco de Código 2.6.1 mostra o esqueleto da classe SomaDoisNumeros, com a definição do bloco do método main e a declaração das 3 variáveis, do tipo double. Perceba a indentação do código.

### Bloco de Código 2.6.1

```
public class SomaDoisNumeros {  
    public static void main(String[] args) {  
        double primeiroValor;  
        double segundoValor;  
        double resultado;  
    }  
}
```

**2.7 Entrada de dados** O Java provê várias classes que podem realizar a entrada de dados. Vamos iniciar pela classe `JOptionPane`, que fornece, entre outros, métodos para entrada e saída. Essa classe está no pacote `javax.swing`, portanto, devemos importar para podermos usar.

O método para entrada de dados é o `showInputDialog`. O Bloco de Código 2.7.1 mostra agora a classe `SomaDoisNumeros`, com a linha de importação da classe `JOptionPane` e a leitura (entrada) dos valores que o usuário digita e são armazenados nas variáveis `primeiroValor` e `segundoValor`.

### Bloco de Código 2.7.1

```
import javax.swing.JOptionPane;  
public class SomaDoisNumeros {  
    public static void main(String[] args) {  
        double primeiroValor;  
        double segundoValor;  
        double resultado;  
        primeiroValor =  
Double.parseDouble(JOptionPane.showInputDialog("Digite o primeiro valor"));  
        segundoValor =  
Double.parseDouble(JOptionPane.showInputDialog("Digite o segundo valor"));  
    }  
}
```

O método `showInputDialog` tem como parâmetro a mensagem que aparece para o usuário (sempre entre aspas). Note também que temos outro método: o `parseDouble` definido pela classe **Double**. Isso é necessário porque o método `showInputDialog` sempre devolve uma `String`, isto é, uma sequência de caracteres, portanto temos que transformar em valor numérico, neste caso em `double`.

**2.8 (Processamento)** Estamos quase lá, a entrada está resolvida, vamos para o processamento. O programa tem por objetivo somar dois números. Utilizamos para isso o operador +. A Tabela 2.8.1 mostra os operadores aritméticos da linguagem Java.

Tabela 2.8.1

Operação	Operador	Exemplo	Resultado
Soma	+	2 + 3	5
Subtração	-	5 - 2	3
Multiplicação	*	6 * 5	30
Divisão inteira	/	5 / 2	2
Divisão real	/	5.0 / 2 ou 5 / 2.0 ou 5d / 2 ou 5 / 2d	2.5
Módulo (Resto de divisão inteira)	%	5 % 2	1

O Bloco de Código 2.8.1 mostra a implementação da fase de processamento.

Bloco de Código 2.8.1

```
import javax.swing.JOptionPane;
public class SomaDoisNumeros {
    public static void main(String[] args) {
        double primeiroValor;
        double segundoValor;
        double resultado;
        primeiroValor =
Double.parseDouble(JOptionPane.showInputDialog("Digite o primeiro valor"));
        segundoValor =
Double.parseDouble(JOptionPane.showInputDialog("Digite o segundo valor"));
        resultado = primeiroValor + segundoValor;

    }
}
```

**2.9 (Saída)** A saída é simples, uma caixa de diálogo para exibir mensagens. O método showMessageDialog se encarrega desta tarefa. Veja o Bloco de Código 2.9.1.

### Bloco de Código 2.9.1

```
import javax.swing.JOptionPane;
public class SomaDoisNumeros {
    public static void main(String[] args) {
        double primeiroValor;
        double segundoValor;
        double resultado;
        primeiroValor =
Double.parseDouble(JOptionPane.showInputDialog("Digite o primeiro valor"));
        segundoValor =
Double.parseDouble(JOptionPane.showInputDialog("Digite o segundo valor"));
        resultado = primeiroValor + segundoValor;
        JOptionPane.showMessageDialog(null, resultado);
    }
}
```

Note que o primeiro parâmetro é a palavra reservada **null**, sobre a qual discutiremos mais adiante. O segundo parâmetro é a expressão que resulta no valor que se deseja exibir.

É possível exibir um simples valor numérico, uma sequência de caracteres e, utilizando o operador de **concatenação**, podemos montar uma sequência textual interessante. Veja o exemplo do Bloco de Código 2.9.2. Repare no comentário que fizemos para que o programa produza a saída uma única vez.

### Bloco de Código 2.9.2

```
import javax.swing.JOptionPane;
public class SomaDoisNumeros {
    public static void main(String[] args) {
        double primeiroValor;
        double segundoValor;
        double resultado;
        primeiroValor =
Double.parseDouble(JOptionPane.showInputDialog("Digite o primeiro valor"));
        segundoValor =
Double.parseDouble(JOptionPane.showInputDialog("Digite o segundo valor"));
        resultado = primeiroValor + segundoValor;
        //JOptionPane.showMessageDialog(null, resultado);
        JOptionPane.showMessageDialog(null, "O resultado é " + resultado);
    }
}
```

**Desafio:** Ajuste o programa para que ele exiba um texto contendo os valores digitados pelo usuário e o resultado final, no seguinte formato:  $a + b = c$ . Por exemplo, se o usuário digitar 2 e 3, seu programa deve exibir: **2 + 3 = 5**.

## Exercícios

1. Ler a cotação do dólar e a quantidade de dólares. Converter para real e mostrar o resultado.
2. Ler 4 números, calcular o quadrado para cada um, somar todos os quadrados e mostrar o resultado.
3. Calcular o pagamento de comissão de vendedores de peças, levando-se em consideração que sua comissão será de 5% do total da venda e que você tem os seguintes dados: preço unitário da peça e quantidade vendida.
4. Ler um valor inteiro e exibir seu antecessor.
5. Ler as dimensões de um retângulo (base e altura), calcular e escrever a área do retângulo.
6. Ler a idade de uma pessoa expressa em anos e exibir expressa em dias (considere que um ano tem 365 dias).
7. Ler a idade de uma pessoa expressa em anos, meses e dias e exibir a idade dessa pessoa expressa apenas em dias. Considerar ano com 365 dias e mês com 30 dias.
8. Ler o número total de eleitores de um município, o número de votos brancos, nulos e válidos. Calcular e escrever o percentual que cada um representa em relação ao total de eleitores.
9. Ler o salário mensal atual de um funcionário e o percentual de reajuste. Calcular e exibir o valor do novo salário.
10. O custo de um carro novo ao consumidor é a soma do custo de fábrica com a porcentagem do distribuidor e dos impostos (aplicados ao custo de fábrica). Supondo que o percentual do distribuidor seja de 28% e os impostos de 45%, ler o custo de fábrica de um carro, calcular e escrever o custo final ao consumidor.
11. Uma revendedora de carros usados paga a seus funcionários vendedores um salário fixo por mês, mais uma comissão também fixa para cada carro vendido e mais 5% do valor das vendas por ele efetuadas. Ler o número de carros por ele vendidos, o valor total de suas vendas, o salário fixo e o valor que ele recebe por carro vendido. Calcular e exibir o salário final do vendedor.