

Algoritmos Básicos: variáveis e entrada e saída de dados

Prof. Ms. Daniel Sant'Anna Consiglieri

(Baseado no material desenvolvido pelo Prof. Bossini e Prof. Machion)

Agenda

1. Objetivos
2. Compilando e executando programas
3. Algoritmos Básicos
4. Exercícios

1.Objetivos

- Estudar algoritmos de sequencia simples (entrada, processamento e saída)
- Implementar estes algoritmos usando a linguagem Java

2. Compilando e executando programas

- Para executar nosso primeiro programa iremos abrir um bloco de notas (Notepad++) e iremos editar um arquivo de texto o conteúdo da *Figura 2.1* e salvá-la como *OlaMundo.java*

O nome do arquivo Java deve ser o nome da única classe pública contida no arquivo.

Figura 2.1 – Exemplo de primeiro programa Java

```
public class OlaMundo {  
    public static void main(String[] args) {  
        System.out.println("Ola Mundo!");  
    }  
}
```

2.Compilando e executando programas

- Na sequencia o código de um programa Java é compilado para uma forma de código intermediária denominado **bytecode**
- O **bytecode** gerado, é interpretado pela Máquina Virtual Java
- Para compilar basta digitar o comando da *Figura 2.2*

Figura 2.2 – Comando para compilar um programa Java

```
javac arquivo.java
```

2.Compilando e executando programas

- Para executar basta digitar o comando mostrado na *Figura 2.3*:

Figura 2.3 – Comando para executar um programa Java

```
java arquivo
```

2. Compilando e executando programas

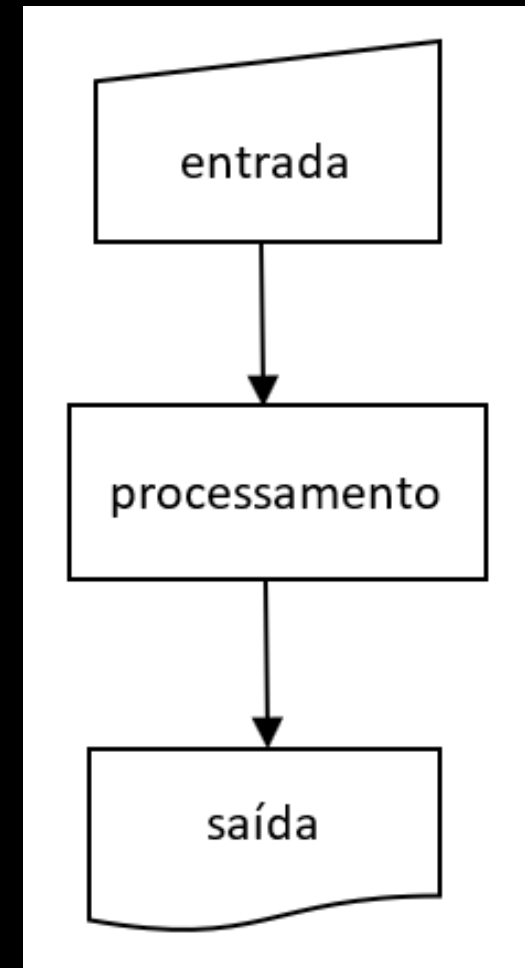
- Observação de boas práticas de programação:
 1. blocos são delimitados por {};
 2. uma boa prática de programação é endentar sempre programas;
 3. classes têm nomes iniciando por letra em caixa alta;
 4. variáveis e métodos têm nome iniciando por letras em caixa baixa;
 5. métodos sempre têm parênteses para argumentos, mesmo que fiquem vazios.
 6. para quaisquer nomes, respeitamos o **CamelCase**

é a denominação em inglês para a prática de escrever as palavras compostas ou frases, sem espaços entre elas, e cada palavra é iniciada com letra maiúscula.

3. Algoritmos Básicos

- Um algoritmo básico ou de sequência simples tem a seguinte estrutura: entrada – processamento – saída. A *Figura 3.1* ilustra um fluxograma para essa estrutura.

Figura 3.1 – Fluxograma para algoritmos básicos



3. Algoritmos Básicos

- 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.

ler 2 números e armazená-los nas variáveis:
primeiroValor e segundoValor

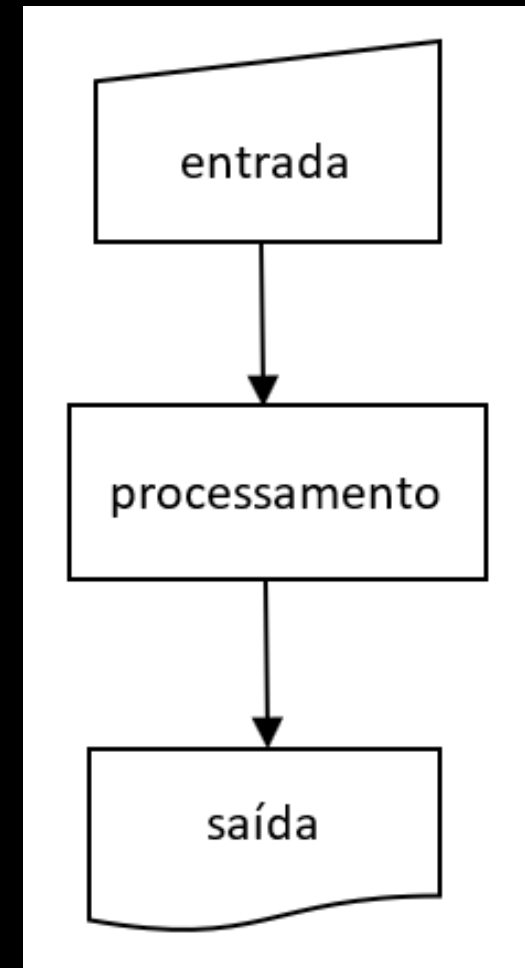
- **Processamento**: é a função do programa. No exemplo é realizar a soma, também armazenando o resultado em uma variável:

resultado = primeiroValor + segundoValor

- **Saída**: exibição do resultado obtido. Para o exemplo, exibir o valor da soma.

exibir resultado

Figura 3.1 – Fluxograma para algoritmos básicos



3.1 Armazenamento dos valores na memória: as variáveis

- 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 válidos.
- No Java temos vários tipos chamados primitivos que permitem trabalhar com vários tipos de informação, vamos ver alguns deles:
 - **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 bit de informação, que pode ser representado pelas palavras false (falso) ou true (verdadeiro).

3.1 Armazenamento dos valores na memória: as variáveis

- A *Figura 3.1.1* 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 endentação do código.

Figura 3.1.1 – Esqueleto da classe SomaDoisNumeros

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

3.2 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.

3.2 Entrada de Dados

- O método para entrada de dados é o `showInputDialog`. A Figura 3.2.1 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`.

Figura 3.2.1 – Esqueleto da classe SomaDoisNumeros usando JOptionPane

```
import javax.swing.JOptionPane;

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

3.2 Entrada de Dados

- a entrada está resolvida, vamos para o processamento. O programa tem por objetivo somar dois números. Utilizamos para isso o operador +. O quadro 3.2.1 mostra os operadores matemáticos mais comuns e a Figura 3.2.2, o código contendo agora a etapa de processamento. Note que foram inseridos também comentários, que te ajudam a lembrar o que está acontecendo em cada trecho de código.

Quadro 3.2.1 – Operadores matemáticos mais comuns

Operador Aritmético	Operação
+	soma
-	subtração
*	multiplicação
/	divisão (depende dos operandos)
%	resto da divisão inteira

3.2 Entrada de Dados

Figura 3.2.2 – Código contendo a etapa do processamento

```
1 import javax.swing.JOptionPane; // é necessário importar a classe, pois não é parte do núcleo
2
3 public class SomaDoisNumeros {
4     public static void main (String[] args) {
5         double primeiroValor, segundoValor, resultado;
6         primeiroValor = Double.parseDouble(JOptionPane.showInputDialog("Digite o primeiro valor:"));
7         segundoValor = Double.parseDouble(JOptionPane.showInputDialog("Digite o segundo valor:"));
8         //processamento
9         resultado = primeiroValor + segundoValor;
10    }
11 }
```

3.2 Entrada de Dados

- A saída é simples, uma caixa de diálogo para exibir mensagens: `showMessageDialog`. Veja o código completo na *Figura 3.2.3*.

Figura 3.2.3 – Código completo

```
1 import javax.swing.JOptionPane; // é necessário importar a classe, pois não é parte do núcleo
2
3 public class SomaDoisNumeros {
4     public static void main (String[] args) {
5         double primeiroValor, segundoValor, resultado;
6         primeiroValor = Double.parseDouble(JOptionPane.showInputDialog("Digite o primeiro valor:"));
7         segundoValor = Double.parseDouble(JOptionPane.showInputDialog("Digite o segundo valor:"));
8         //processamento
9         resultado = primeiroValor + segundoValor;
10        //saída
11        JOptionPane.showMessageDialog(null, "soma = " + resultado);
12    }
13 }
```


3.2 Entrada de Dados

- Note que o primeiro parâmetro é a palavra reservada **null**, a qual discutiremos mais adiante. O segundo parâmetro é a sentença que se deseja exibir; é possível exibir uma mensagem simples, sempre entre aspas ou uma mensagem concatenada a valores de variáveis, utilizando-se o operador + (linha 11).
- Para fixar o conteúdo refaça o programa anterior e verifique se entendeu todos as linhas de códigos

4.Exercícios: Desenvolver um programa em Java para resolver os seguintes problemas:

- 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.

4.Exercícios: Desenvolver um programa em Java para resolver os seguintes problemas:

- 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.

4.Exercícios: Desenvolver um programa em Java para resolver os seguintes problemas:

- 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.