

# Aula 05

## Funções

Rogério Pereira Junior

rogeriopereirajunior@gmail.com

7 de maio de 2024

# Funções

## Funções

- é um bloco de código que executa uma tarefa específica quando chamado ou invocado.
- Ela pode aceitar argumentos como entrada, realizar operações com esses argumentos e retornar um resultado.
- Funções podem ou não retornar uma saída
- Representam um processamento que possui um significado
  - `Math.sqrt(double)`
  - `System.out.println(string)`

## Finalidade

As funções são usadas para modularizar o código, o que significa que você pode dividir um programa em partes menores e mais gerenciáveis, cada uma responsável por uma tarefa específica.

# Funções

- Em Java, uma função é geralmente chamada de "método".
- Métodos são blocos de código que realizam uma tarefa específica e podem ser invocados (chamados) de outros lugares no programa.

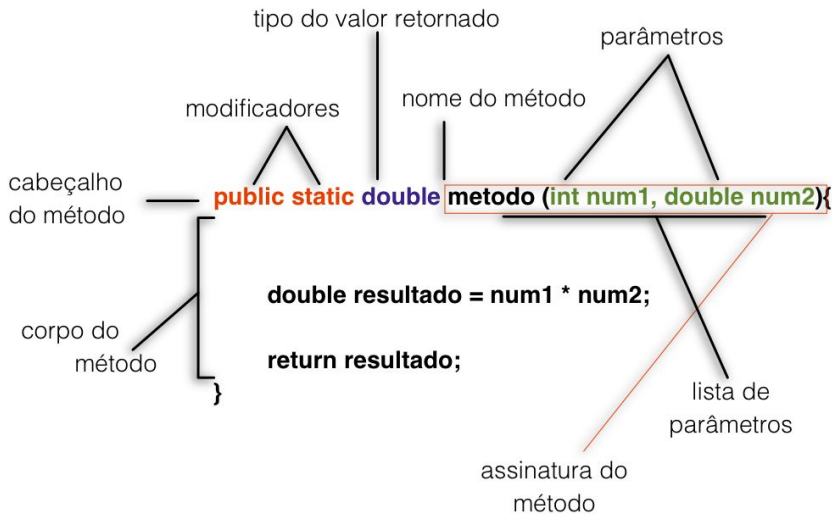
```
// Método principal (ponto de entrada do programa)
public static void main(String[] args) {
    // Chamando a função e armazenando o resultado em uma variável
    int resultado = soma(5, 3);

    // Imprimindo o resultado
    System.out.println("A soma é: " + resultado);
}

// Definindo a função (método) para somar dois números inteiros
public static int soma(int a, int b) {
    int resultado = a + b;
    return resultado; // Retornando o resultado da soma
}
```

- **main** é o método principal, que é o ponto de entrada do programa.
- O método **main** chama o método **soma** e imprime o resultado.
- A palavra **void** pode ser utilizada caso o método não retorne nenhum dado.
- Dentro dos parênteses após o nome do método nós definimos os **parâmetros** que o método necessita para realizar suas operações/tarefas

# Funções



- Modificadores de acesso: especificam a visibilidade de classes, métodos, atributos e outros membros dentro de um programa. Eles controlam quais partes do código podem acessar ou modificar determinados elementos.

# Funções

- Fazer um programa para ler três números inteiros e mostrar na tela o maior deles

```
System.out.println("Entre com os 3 números:");
int n1 = sc.nextInt();
int n2 = sc.nextInt();
int n3 = sc.nextInt();
if (n1 > n2 && n1 > n3) {
    System.out.println("Maior = " + n1);
} else if (n2 > n3) {
    System.out.println("Maior = " + n2);
} else {
    System.out.println("Maior = " + n3);
}
```

- Vamos transformar em funções

```
public static int maior(int x, int y, int z) {
    int aux;
    if (x > y && x > z) {
        aux = x;
    } else if (y > z) {
        aux = y;
    } else {
        aux = z;
    }
    return aux; }
public static void mostrarResultado(int valor) {
    System.out.println("Maior = " + valor);
}
```

# Funções - Exemplo

```
public static void main(String[] args) {  
    // Chamando a função soma e imprimindo o resultado  
    int resultadoSoma = soma(5, 3);  
    System.out.println("Resultado da soma: " + resultadoSoma);  
  
    // Chamando a função que verifica se um número é par  
    int numero = 7;  
    if (ehPar(numero)) {  
        System.out.println(numero + " é par.");  
    } else {  
        System.out.println(numero + " não é par.");  
    }  
}  
  
// Função que calcula a soma de dois números inteiros  
public static int soma(int a, int b) {  
    return a + b;  
}  
  
// Função que verifica se um número é par  
public static boolean ehPar(int numero) {  
    return numero % 2 == 0;  
}  
}
```

# Funções

```
// Solicitando informações sobre a primeira pessoa
System.out.println("Informe os dados da primeira pessoa:");
System.out.print("Nome: ");
String nome1 = sc.nextLine();
System.out.print("Idade: ");
int idade1 = sc.nextInt();
sc.nextLine(); // Limpar o buffer do teclado
// Solicitando informações sobre a segunda pessoa
System.out.println("Informe os dados da segunda pessoa:");
System.out.print("Nome: ");
String nome2 = sc.nextLine();
System.out.print("Idade: ");
int idade2 = sc.nextInt();
// Chamando a função para determinar a pessoa mais velha
String pessoaMaisVelha = determinarPessoaMaisVelha(nome1, idade1, nome2, idade2);
System.out.println("Pessoa mais velha: " + pessoaMaisVelha);
sc.close();
}

// Função que determina e retorna o nome da pessoa mais velha
public static String determinarPessoaMaisVelha(String nome1, int idade1, String nome2, int idade2) {
    if (idade1 > idade2) {
        return nome1;
    } else if (idade2 > idade1) {
        return nome2;
    } else {
        return "As duas pessoas têm a mesma idade.";
    }
}
```

## Funções - Exemplo juros

```
double capital = 1000; // Capital inicial
double taxa = 0.05;    // Taxa de juros (5%)
int periodo = 3;       // Período em anos

// Calculando juros simples
double jurosSimples = calcularJurosSimples(capital, taxa, periodo);
System.out.println("Juros simples após " + periodo + " anos: " + jurosSimples);

// Calculando montante (capital + juros) com juros simples
double montanteSimples = capital + jurosSimples;
System.out.println("Montante com juros simples após " + periodo + " anos: " + montanteSimples);
}

// Função para calcular juros simples
public static double calcularJurosSimples(double capital, double taxa, int periodo) {
    return capital * taxa * periodo;
}

// Função para calcular montante com juros compostos
public static double calcularMontanteComposto(double capital, double taxa, int periodo) {
    return capital * Math.pow(1 + taxa, periodo);
}
```



# Funções - Módulos

- Depois de criarmos várias funções, os programas ficaram muito grandes. Precisamos armazenar nossas funções em outros arquivos e, de alguma forma usá-las, sem precisar reescrevê-las, ou pior, copiar e colar.
- Classe **Operacoes**

```
public class Operacoes {  
    public static int somar(int num1, int num2) {  
        int soma = num1 + num2;  
        return soma;  
    }  
    public static int subtrair(int num1, int num2) {  
        int subtrair = num1 - num2;  
        return subtrair;  
    }  
    public static int multiplicar(int num1, int num2) {  
        int produto = num1 * num2;  
        return produto;  
    }  
    public static double dividir(int num1, int num2) {  
        if (num2 != 0) {  
            return (double) num1/num2;  
        }  
        return 0;  
    }  
}
```

- Chamamos isto de **Modulo**

# Funções - Módulos

Em um outra classe dentro da função **main** chamamos o módulo da seguinte forma

```
public class Programa02 {  
  
    public static void main(String[] args) {  
  
        int resultado = Operacoes.multiplicar(2, 4);  
        System.out.println("Resultado: " + resultado);  
  
    }  
  
}
```