



# Modularização na Linguagem Java

Prof. Me. Rober Marcene Rosi  
Unidade de Engenharia e Computação - FAESA

# Vantagens

## Modularização

- ☐ Redução de complexidade
- ☐ Identificação de erros facilitada
- ☐ Reutilização
- ☐ Manutenção mais simples
- ☐ Expansibilidade

# Modularização

**Programação Estruturada  
x  
Programação Orientada a  
Objetos**



# Exemplos

## Programação Estruturada

```
1. class CalculadoraProcedural {
2.     public static void main (String [ ] args) {
3.         Scanner ler = new Scanner(System.in);
4.         double num1, num2;
5.         System.out.println("Digite o primeiro valor: ");
6.         num1 = ler.nextDouble();
7.         System.out.println("Digite o segundo valor");
8.         num2 = ler.nextDouble();
9.         multiplicarNumeros (num1, num2);
10.        System.exit(0);
11.    }
12.    static void somarNumeros (double a, double b) {
13.        int res;
14.        res = a + b;
15.        System.out.println("Resultado Soma: " + res);
16.    }
```

# Exemplos

## Programação Estruturada

```
17. static void subtrairNumeros (double a, double b) {
18.     int res;
19.     res = a - b;
20.     System.out.println("Resultado Subtração: " + res);
21. }
17. static void multiplicarNumeros (double a, double b) {
18.     int res;
19.     res = a * b;
20.     System.out.println("Resultado Multiplicação: " + res);
21. }
17. static void dividirNumeros (double a, double b) {
18.     int res;
19.     res = a / b;
20.     System.out.println("Resultado Divisão: " + res);
21. }
22. }
```



# Exemplos

## Programação Orientada a Objetos

```
1. class CalculadoraOO {  
2.     public static void main (String [ ] args) {  
3.         Scanner ler = new Scanner(System.in);  
4.         Multiplica res = new Multiplica();  
5.         double num1, num2;  
6.         System.out.println("Digite o primeiro valor: ");  
7.         num1 = ler.nextDouble();  
8.         System.out.println("Digite o segundo valor");  
9.         num2 = ler.nextDouble();  
10.        res.multiplicaNumeros (num1, num2);  
11.        ler.close();  
12.        System.exit(0);  
13.    }  
14. }
```

# Exemplos

## Programação Estruturada

```
1. public class Soma {  
2.     public void somarNumeros (double a, double b) {  
3.         int res;  
4.         res = a + b;  
5.         System.out.println("Resultado Soma: " + res);  
6.     }  
7, }
```

```
1. public class Subtrai {  
2.     public void subtrairNumeros (double a, double b) {  
3.         int res;  
4.         res = a + b;  
5.         System.out.println("Resultado Subtração: " + res);  
6.     }  
7, }
```

# Exemplos

## Programação Estruturada

```
1. public class Multiplica {  
2.     public void multiplicarNumeros (double a, double b) {  
3.         int res;  
4.         res = a * b;  
5.         System.out.println("Resultado Multiplicação: " + res);  
6.     }  
7, }
```

```
1. public class Divide {  
2.     public void dividirNumeros (double a, double b) {  
3.         int res;  
4.         res = a / b;  
5.         System.out.println("Resultado Divisão: " + res);  
6.     }  
7, }
```



# Procedimento

Exemplo

```
static void calcularAdicao (double v1, double v2){  
    double res;  
    res = v1 + v2;  
    System.out.println( "Soma = " + res);  
}
```

# Função

Exemplo

```
static double calcularAdicao (double v1, double v2){  
    return (v1 + v2);  
}
```

# Exemplos

## Precedimento

```
1. class TestaSoma {  
2.     public static void main (String [ ] args) {  
3.         Scanner ler = new Scanner(System.in);  
4.         double num1, num2;  
5.         System.out.println("Digite o primeiro valor: ");  
6.         num1 = ler.nextDouble();  
7.         System.out.println("Digite o segundo valor");  
8.         num2 = ler.nextDouble();  
9.         calcularAdicao (num1, num2);  
10.        ler.close();  
11.        System.exit(0);  
12.    }  
13. }
```

# Exemplos

## Função

```
1. class TestaSoma {  
2.     public static void main (String [ ] args) {  
3.         Scanner ler = new Scanner(System.in);  
4.         double num1, num2, resultado;  
5.         System.out.println("Digite o primeiro valor: ");  
6.         num1 = ler.nextDouble();  
7.         System.out.println("Digite o segundo valor");  
8.         num2 = ler.nextDouble();  
9.         Resultado = calcularAdicao (num1, num2);  
10.        System.out.println("Resultado da soma: " + resultado);  
11.        ler.close();  
12.        System.exit(0);  
13.    }  
14. }
```

# Procedimento x Função

Comparação entre Procedimento e Função

## Procedimento

*Procedimento*

*ModAdicao(v1, v2: real)*

*Var*

*res: real*

*Início*

*res*  $\leftarrow$  *v1* + *v2*

*Mostrar* (*res*)

*Fim.*

## Função

*Função*

*ModAdicao(v1, v2: real): real*

*Var*

*res: real*

*Início*

*res*  $\leftarrow$  *v1* + *v2*

*retornar* (*res*)

*Fim.*



Perguntas?

A close-up photograph of a black computer keyboard. A gold-colored key with an ornate, decorative head is resting on the Enter key. The Enter key is labeled "Enter" and has a white arrow pointing to the right. Other keys visible include "Insert", "Delete", and "End". The background is dark and slightly blurred, with some keys like "]" and "[" visible. The overall composition is centered and uses a shallow depth of field.

Obrigado