

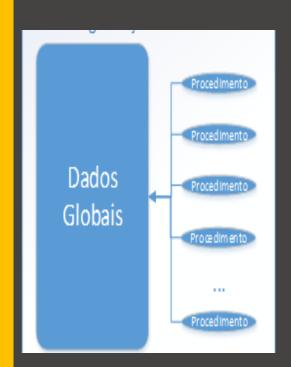
### Vantagens

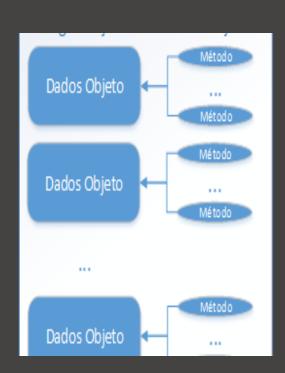
Modularização

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

### Modularização

Programação Estruturada X Programação Orientada a Objetos





```
class CalculadoraProcedural {
                    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.
```

```
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.
```

#### Programação Orientada a Objetos

```
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.
```

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

```
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

```
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);
}
```

#### Precedimento

```
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();
                               System.exit(0);
11.
12.
13.
```

Função

```
1. class TestaSoma {
2.
          public static void main (String [ ] args) {
                    Scanner ler = new Scanner(System.in);
                    double num1, num2, resultado;
                     System.out.println("Digite o primeiro valor: ");
                     num1 = ler.nextDouble();
6.
                    System.out.println("Digite o segundo valor");
                    num2 = ler.nextDouble();
                    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**

### Função

```
Procedimento
 ModAdicao(v1, v2: real)
Var
 res: real
Início
 res \leftarrow v1 + v2
 Mostrar (res)
Fim.
```

```
Função
ModAdicao(v1, v2: real): real
Var
res: real
Início
res \leftarrow v1 + v2
retornar (res)
Fim.
```





Obrigado