

# Modularização

## Procedimentos e Funções

# Procedimentos e Funções

---

- **Objetivo:**

Procedimentos, funções e parâmetros: recursos utilizados para tornar os algoritmos mais eficientes e possibilitar a reutilização de código;

Uso de algumas rotinas em vários programas, inclusive com objetivos diferentes;

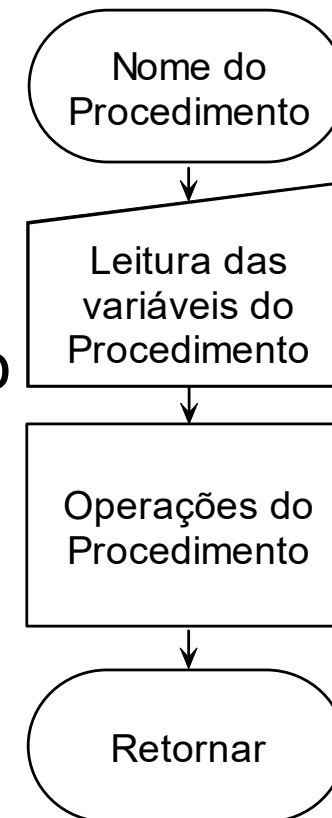
Escopo de variáveis;

Parâmetros;

Passagem de parâmetros.

# Procedimentos

- Um procedimento (procedure), também conhecido como sub-rotina, é um conjunto de instruções que realiza uma determinada tarefa;
- Um algoritmo de procedimento é criado da mesma maneira que outro algoritmo qualquer: Deve ser identificado, possui variáveis, operações e até funções.



# Procedimento

---

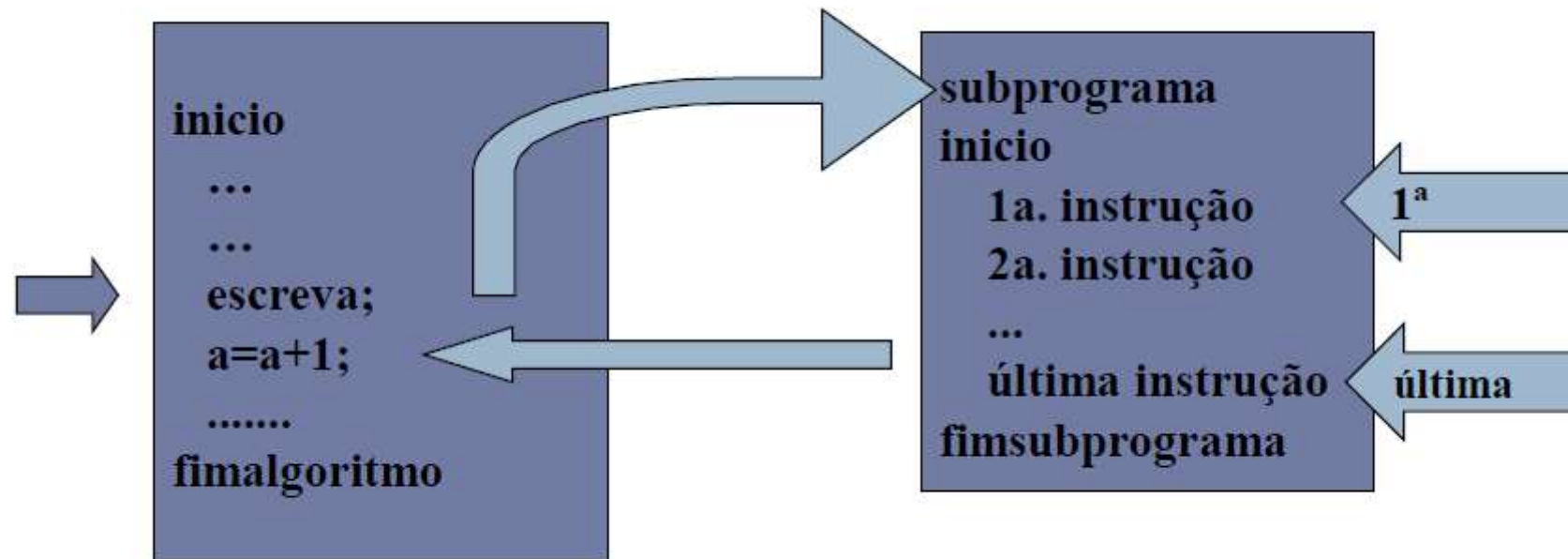
- Um procedimento é um bloco de programa, contendo início e fim e será identificado por um nome, através do qual será referenciado em todo o programa principal.
- Quando uma sub-rotina é chamada por um programa, ela é executada e ao seu término, o controle do processamento retorna para a 1ª linha de instrução.

# Códigos para Códigos

---

- **Vantagens na utilização de procedimentos e funções.**
  - Melhorar a clareza e compreensão do programa;
  - Diminui o tamanho do código;
  - Facilita a alteração;
  - Diminui a quantidade de erros;
  - Diminui a complexidade;
  - Facilita o gerenciamento;
  - Independência;
  - Reutilização;

# Códigos para Códigos



# Códigos para Códigos - Função (Function)

---

- A estrutura de uma função é muito parecida com um procedimento.
- Pode-se imaginar que uma função é um procedimento com características especiais **quanto ao retorno de valores.**
- A **função** é um subprograma que pode agir sobre os dados e **retornar um (OU VÁRIOS)** valor para o programa principal.

# Códigos para Códigos - Função (Function)

---

- Uma característica que distingue uma função de um procedimento é que a função pode ser impressa, atribuída ou participar de cálculos como se fosse uma variável qualquer.
- Isso por que ela tem um valor retorno.



# Códigos para Códigos - Função (Function)

---

## Sintaxe:

**funcao** <identificador> ([var]<parâmetros>) <tipo de retorno>

var <declare as variáveis>

inicio

<comandos>

**retorne** <variável de retorno>

**fimfuncao**

# Códigos para Códigos - Função (Function)

---

Algoritmo "Soma"

var

a, b, result: inteiro

**funcao ImprimeSoma(x:inteiro; y:inteiro):inteiro**

**Var**

**soma: inteiro**

**inicio**

**soma <- x + y**

**retorne soma**

**fimfuncao**

**{Programa Principal}**

inicio

Escreva("Entre com o primeiro valor: " )

Leia(a)

Escreva("Entre com o segundo valor: " )

Leia(b)

result <- **ImprimeSoma(a,b)**

Escreva("A soma dos valores é igual a: ", result)

fimalgoritmo

# Códigos para Códigos - Função (Function)

---

Algoritmo "Soma"

var

a, b, result: inteiro

**funcao ImprimeSoma(x:inteiro; y:inteiro):inteiro**

**Var**

**soma: inteiro**

**inicio**

**soma <- x + y**

**retorne soma**

**fimfuncao**

**{Programa Principal}**

inicio

Escreva("Entre com o primeiro valor: " )

Leia(a)

Escreva("Entre com o segundo valor: " )

Leia(b)

result <- **ImprimeSoma(a,b)**

Escreva("A soma dos valores é igual a: ", result)

fimalgoritmo

# Códigos para Códigos - Função (Function)

---

Quando declaramos as variáveis, nós podemos fazê-las;

- **Local** - declaradas dentro de um procedimento ou função - só têm validade dentro do escopo ao qual foram declaradas.
- **Global** - são as variáveis do programa principal. São acessíveis a todos os subprogramas e ao programa principal.

Obs: Parâmetros são variáveis locais.

# Códigos para Códigos - Função (Function)

Algoritmo "Soma"

var

a, b, result: inteiro

funcao ImprimeSoma(x:inteiro; y:inteiro):inteiro

Var

soma: inteiro

inicio

soma <- x + y

retorne soma

fimfuncao

{Programa Principal}

inicio

Escreva("Entre com o primeiro valor: " )

Leia(a)

Escreva("Entre com o segundo valor: " )

Leia(b)

result <- ImprimeSoma(a,b)

Escreva("A soma dos valores é igual a: ", result)

fimalgoritmo

**Variável Global** (funciona em  
TODO código principal)



**Variável Local** (funciona dentro  
da estrutura, no caso a função)

# Códigos para Códigos - Procedimentos (procedure)

---

É um subprograma que **não retorna nenhum valor**

Deve ser definido após a declaração das variáveis.

É ativado ao ser chamado no programa principal.

Pode ou não ter parâmetros.

# Exemplo – Cálculo de Imposto

```
1  import java.util.Scanner;
2  import java.io.*;
3  import java.text.DecimalFormat;
4  public class Aula5F_1 {
5      public static void main(String[] args) {
6          double salario, imposto;
7          Scanner ler = new
8              Scanner(System.in);
9          DecimalFormat deci = new DecimalFormat("0.00");
10
11          System.out.println("Digite seu salario: ");
12          salario = ler.nextDouble();
13          imposto=CalculaImposto(salario);
14          System.out.println("Você pagará em impostos: R$"+ deci.format(imposto));
15      }
16      public static double CalculaImposto(Double x){
17          double desconto;
18          desconto=x*0.14;
19          return desconto;
20      }
21  }
22  }
23  }
```

# Códigos para Códigos - Função (Function)

Você deve desenvolver um sistema de contabilidade. Porém você identificou que há um processo que se repete diversas vezes, no caso o cálculo de um imposto, dada uma alíquota. E isso te deixa meio #xatiado por que são vários impostos e a avaliação é de 12 meses (para cada mês, há o cálculo de INSS, Simples, CREA, ICMS, ISS, e cada um deles deve ser descrito quando descontado). Porém você programa **pra caramba**..... então faça um código que receba 12 faturamentos, e que calcule ( e demonstre na tela) o valor descontado de cada imposto para o respectivo mês, com um código enxuto e com os cálculos automatizados de impostos.

Seguem as alíquotas:

INSS: 3%

Simples: 2,5%

CREA: 1,45%

ICMS: 2%

ISS: 1,67%

```
-----
Digite seu faturamento no mês: 1
120000,00
Você pagará de INSS no mês 1: R$3600,00
Você pagará de Simples no mês 1: R$3000,00
Você pagará de CREA no mês 1: R$1740,00
Você pagará de ICMS no mês 1: R$2400,00
Você pagará de ISS no mês 1: R$2004,00
-----
```



# Códigos para Códigos - Função (Function)

Boooooooooooooooooooooooooora praticar

Você deve desenvolver um sistema de contabilidade. Porém você identificou que há um processo que se repete diversas vezes, no caso o calculo de um imposto, dada uma alíquota. E isso te deixa meio #xtiado por que são vários impostos e a avaliação é de 12 meses (para cada mês, há o cálculo de INSS, Simples, CREA, ICMS, ISS, e cada um deles deve ser descrito quando descontado). Porém você programa **pra caramba**..... então faça um código que receba 12 faturamentos, e que calcule ( e demonstre na tela) o valor descontado de cada imposto para o respectivo mês, com um código enxuto e com os cálculos automatizados de impostos.

Seguem as alíquotas:

INSS: 3%  
Simples: 2,5%  
CREA: 1,45%  
ICMS: 2%  
ISS: 1,67%

```
5 package aula5f_2;
6 import java.util.Scanner;
7 import java.io.*;
8 import java.text.DecimalFormat;
9 /**
10  *
11  * @author flaba
12  */
13 public class Aula5F_2 {
14
15     /**
16      * @param args the command line arguments
17      */
18     public static void main(String[] args) {
19         // TODO code application logic here
20         double faturamento, inss, simples, crea, icms, iss;
21         Scanner ler = new
22             Scanner(System.in);
23         DecimalFormat deci = new DecimalFormat("0.00");
24
25         for(int i=0; i<12; i++){
26             System.out.println("Digite seu faturamento no mês "+(i+1));
27             faturamento = ler.nextDouble();
28
29             inss=CalculaImposto(faturamento,0.03);
30             System.out.println("Você pagará de INSS no mês "+(i+1)+" R$"+ deci.format(inss));
31             simples=CalculaImposto(faturamento,0.025);
32             System.out.println("Você pagará de Simples no mês "+(i+1)+" R$"+ deci.format(simples));
33             crea=CalculaImposto(faturamento,0.0145);
34             System.out.println("Você pagará de CREA no mês "+(i+1)+" R$"+ deci.format(crea));
35             icms=CalculaImposto(faturamento,0.02);
36             System.out.println("Você pagará de ICMS no mês "+(i+1)+" R$"+ deci.format(icms));
37             iss=CalculaImposto(faturamento,0.0167);
38             System.out.println("Você pagará de ISS no mês "+(i+1)+" R$"+ deci.format(iss));
39         }
40     }
41     public static double CalculaImposto(Double x, Double y){
42         double desconto;
43         desconto=x*y;
44         return desconto;
45     }
46 }
47 }
```

# Códigos para Códigos - Procedimentos (procedure)

---

Algoritmo "Ola"

var

nome: literal

procedimento elogio

inicio

Escreva("Bom dia " , nome , " seja bem  
vindo!")

fimprocedimento

//Programa Principal

inicio

Escreva(" Informe o seu nome: ")

Leia(nome)

elogio

fimalgoritmo

# Códigos para Códigos - Procedimentos (procedure)

---

**Identificador:** nome do procedimento também tratado como identificador.

**Passagem de parâmetros por referência:** utiliza-se a construção VAR antes dos identificadores para indicar a passagem por referência. Os identificadores são separados por vírgula.

**Parâmetros:** Entre um mesmo tipo de dados são separados por vírgula. Entre tipos de dados a separação é feita com ponto-e-vírgulas ';'.

# Códigos para Códigos - Procedimentos (procedure)

---

Toda vez que precisarmos chamar o procedimento devemos chamar por esse nome/identificador.

As declarações de variáveis dentro do bloco do procedimento, são opcionais.

O nome do procedimento obedece as mesmas regras de nomenclatura das variáveis.

# Códigos para Códigos - Funções e Procedimentos

```
1 def dez():                # Função sem parâmetro
2     return 10
3
4 def ex_soma(num1, num2):   # Função com parâmetro
5     s = num1 + num2
6     return s
7
8 def exibir(msg, valor):    # Procedimento ou sub-rotina
9     print(msg, valor)
10
11 numero1 = dez()           # Função irá retornar o número 10
12
13 exibir('O valor armazenado em numero1 =', numero1) # Chama a subrotina para exibir a mensagem e o valor armazenado em numero1
14
15 numero2 = dez()**2        # O número 10 retornado da função será elevado ao quadrado
16
17 exibir('O valor armazenado em numero2 =', numero2) # Chama a subrotina para exibir a mensagem e o valor armazenado em numero2
18
19 soma = ex_soma(numero1, numero2) # Chama a função para realizar a soma de 10 (numero 1) e 100 (numero2)
20
21 exibir("A soma dos números é igual a", soma) # Chama a subrotina para exibir a mensagem e o resultado da soma
22
```

# Códigos para Códigos - Funções e Procedimentos

Exemplo:

```
5 package aula5f_3;
6 import java.util.Scanner;
7 import java.io.*;
8 /**
9  *
10  * @author flaba
11  */
12 public class Aula5F_3 {
13
14     /**
15      * @param args the command line arguments
16      */
17     public static void ConectaBanco() {
18         System.out.println("Conectando ao Bando de dados");
19         System.out.println("Obtendo Dados do Servidor");
20         System.out.println("Conectado");
21     }
22     public static void DesconectaBanco() {
23         System.out.println("Desconectando ao Bando de dados");
24         System.out.println("Enviando Dados do Servidor");
25         System.out.println("Desconectado");
26     }
27     public static void main(String[] args) {
28         // TODO code application logic here
29         System.out.println("Digite Seu Comando:");
30         System.out.println("Salvando seu comando na base de dados");
31         ConectaBanco();
32         DesconectaBanco();
33     }
34 }
35
```

aula5f\_3.Aula5F\_3 main

Output - Aula5F\_3 (run) x

```
run:
Digite Seu Comando:
Salvando seu comando na base de dados
Conectando ao Bando de dados
Obtendo Dados do Servidor
Conectado
Desconectando ao Bando de dados
Enviando Dados do Servidor
Desconectado
BUILD SUCCESSFUL (total time: 1 second)
```

# Códigos para Códigos - Funções e Procedimentos

---

Pratiquemos: Desenvolva um software com a seguinte especificação:

Em uma empresa de logística há o processo de embarque de cargas. O software pergunta ao seu motorista qual a carga máxima de seu caminhão. respondida a carga, o software pergunta se ele gostaria de cadastrar seu caminhão para serviços.

Caso ele digite “s”, de sim, o sistema executa um processo de conexão com banco de dados e registro do caminhão que ele demonstra através de mensagens “Conectando ao Banco de Dados”, “Caminhão Cadastrado!”. Depois, ele realiza um cálculo de transbordo, onde ele calcula a carga máxima que o caminhão poderá transportar, retirando uma margem de 5% de segurança. Essa carga efetiva de transporte deve ser multiplicada por uma taxa de R\$120,54, que é o preço por tonelada transbordada (imprima o valor a ser pago).

Caso ele digite “n” , de não, o sistema deve agradecer e finalizar.

# Códigos para Códigos - Funções e Procedimentos

Pratiquemos: Desenvolva um software com a seguinte especificação:

Em uma empresa de logística há o processo de embarque de cargas. O software pergunta ao seu motorista qual a carga máxima de seu caminhão. respondida a carga, o software pergunta se ele gostaria de cadastrar seu caminhão para serviços.

Caso ele digite "s", de sim, o sistema executa um processo de conexão com banco de dados e registro do caminhão que ele demonstra através de mensagens "Conectando ao Banco de Dados", "Caminhão Cadastrado!". Depois, ele realiza um cálculo de transbordo, onde ele calcula a carga máxima que o caminhão poderá transportar, retirando uma margem de 5% de segurança. Essa carga efetiva de transporte deve ser multiplicada por uma taxa de R\$120,54, que é o preço por tonelada transbordada (imprima o valor a ser pago).

Caso ele digite "n" , de não, o sistema deve agradecer e finalizar.

```
5 package aula5f_4;
6 import java.util.Scanner;
7 import java.io.*;
8 /**
9  *
10  * @author flaba
11  */
12 public class Aula5F_4 {
13     /**
14      * @param args the command line arguments
15      */
16     public static void ConectaBanco(){
17         System.out.println("Conectando ao Banco de dados");
18         System.out.println("Caminhão Cadastrado");
19     }
20     public static double CalculaTransbordo(Double x){
21         double seguranca;
22         seguranca=x*0.95;
23         return seguranca;
24     }
25     public static void main(String[] args) {
26         // TODO code application logic here
27         double carga, cargasegura, transporte;
28         String servico;
29         Scanner ler = new
30             Scanner(System.in);
31
32         System.out.println("Qual a Carga máxima de seu caminhão? ");
33         carga = ler.nextDouble();
34
35         System.out.println("Gostaria de Cadastrar seu Caminhão para serviços? ");
36         servico = ler.next();
37
38         if(servico.equals("s")){
39             ConectaBanco();
40             cargasegura=CalculaTransbordo(carga);
41             transporte=cargasegura*120.54;
42             System.out.println("Frete a ser pago: "+transporte);
43         }else{
44             System.out.println("Obrigado! Finalizando... ");
45         }
46     }
47 }
```



# Vamos praticar

---

Fazer um programa para obter três notas e calcular e exibir a sua média, usando uma função para o cálculo da média.

Elabore um procedimento e uma função que receba um número como entrada e retorne verdade caso este número for par, e falso caso contrário.

---

Escrever um algoritmo que leia uma quantidade desconhecida de números e conte quantos deles estão nos seguintes intervalos:  $[0, 25]$ ,  $[26, 50]$ ,  $[51, 75]$  e  $[76, 100]$ . A entrada de dados deve terminar quando for lido um número negativo.