Programação I

Métodos estáticos

Samuel da Silva Feitosa

Aula 4a

Métodos Estáticos

Introdução

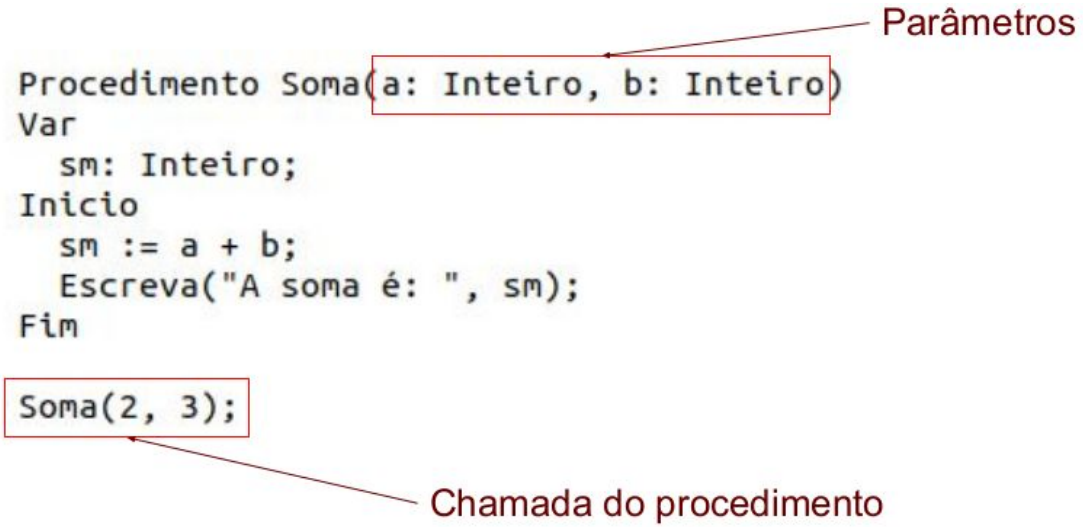
- A modularização de código é algo essencial no desenvolvimento de software atualmente.
  - Procedimentos e funções são conceitos utilizados para este fim.
- A linguagem Java implementa ambos os conceitos através de métodos.
  - É possível separar as implementações em múltiplos arquivos.

O que já sabemos?

- É comum dividir o problema / algoritmo em diversos pedaços, cada qual com sua funcionalidade.
  - Daí vem o nome **modularização** ou **programação estruturada**.
- Já conhecemos os conceitos de **procedimentos** e **funções**.
  - Ambos são trechos de código separados da linha de execução principal, que podem ser **chamados** quantas vezes for necessário.

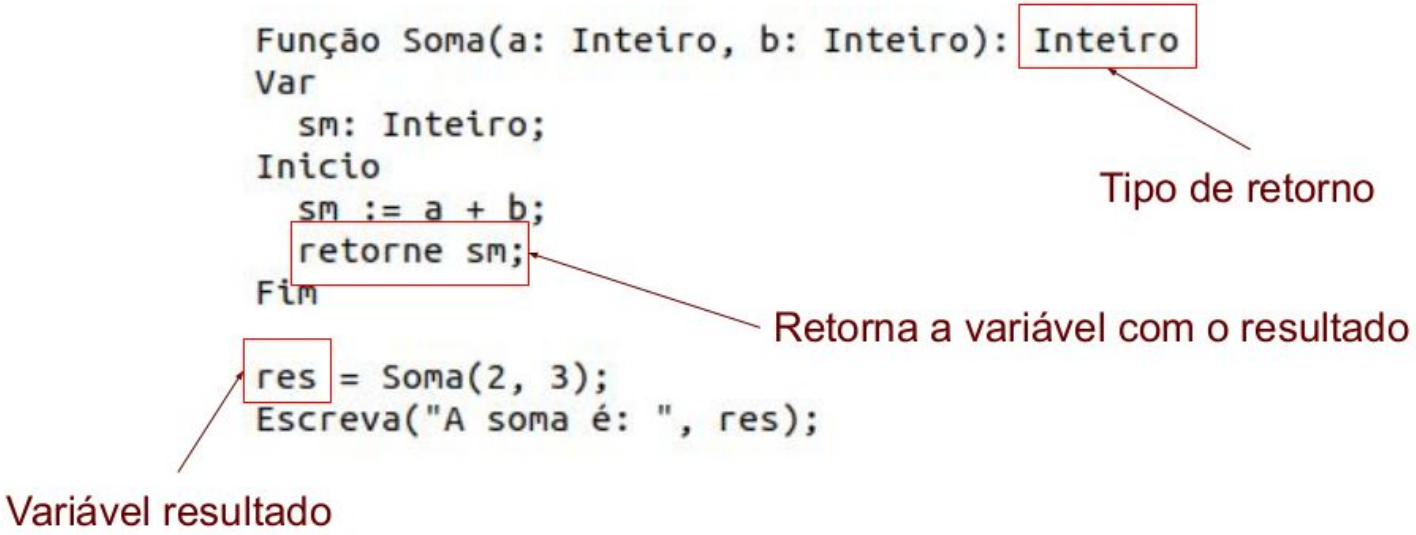
Procedimentos

- Um procedimento é uma sub-rotina que realiza uma operação e não retorna nenhum valor.



Funções

- Uma função é uma sub-rotina que realiza alguma operação e retorna algum valor.



Parâmetros e Escopo

- Informações e variáveis podem ser **passadas por parâmetro** através da chamada de funções ou procedimentos.
- Variáveis criadas **dentro de blocos** tem **escopo local**, ou seja, valem apenas dentro do bloco em que foram criadas.
    - Outras variáveis não podem ser utilizadas, a menos que sejam variáveis globais.

Vantagens da Modularização

- **Dividir** e **estruturar** um algoritmo em partes logicamente coerentes.
- Facilidade de testar trechos em separado.
- Evitar repetição do código-fonte.
- Maior legibilidade de código.

Uso na Linguagem Java

- A linguagem Java não possui conceitos separados para procedimentos e funções.
- Para implementar ambas as funcionalidades são utilizados os **métodos**.
- De forma similar, **métodos** são trechos de código que permitem realizar ações ou transformações sobre valores.

Métodos sem retorno

- Um método sem retorno possui o mesmo comportamento de um procedimento.
  - Para indicar que o método não tem retorno usamos a palavra-chave **void**.

```
static void soma(int a, int b) {
    int sm = a + b;
    System.out.println("A soma é: " + sm);
}
public static void main(String[] args) {
    soma(2, 3);
}
```

Métodos com retorno

- Um método com retorno possui a mesma funcionalidade de uma função.

```
static int soma(int a, int b) {
    int sm = a + b;
    return sm;
}

public static void main(String[] args) {
    int res = soma(2, 3);
    System.out.println("A soma é: " + res);
}
```

Múltiplos arquivos

- Métodos em Java podem estar dispostos em múltiplos arquivos.
  - Essa divisão facilita a administração de código, deixando cada arquivo responsável por uma funcionalidade específica do sistema.
  - Permite também a reutilização de código, onde é possível realizar o compartilhamento apenas dos arquivos (bibliotecas) necessários para outros projetos.
- Vejamos alguns exemplos.

Classe java.lang.Math

- Classe que contém métodos estáticos para a realização de cálculos numéricos.
  - E/PI: representam as constantes de Euler (e) e PI.
  - acos / asin / atan, cos / sin / tan: relacionados ao coseno, seno e tangente.
  - ceil / floor / rint / round: teto, piso, arredondamentos.
  - sqrt / cbrt / pow: raiz quadrada, cúbica e potência.
  - log / log10: logaritmos.
  - abs, max / min: valor absoluto, máximo e mínimo.
  - random: retorna um valor aleatório.

Considerações Finais

- Métodos em Java seguindo a semântica de procedimentos e funções.
  - Vimos como modularizar um programa através de bibliotecas, ou arquivos de código fonte, separados do programa principal.
- Vimos como fazer a **depuração** (**debug**) de código Java.