



Algoritmos e Programação I

Funções

Prof. Alexandre Mignon

Programação Modular

- Consiste em dividir o programa em subprogramas.
- Subprograma é um trecho de programa que contém início e fim, e executa um determinado conjunto de instruções.
- Todo subprograma possui um nome e uma estrutura.
- Podem ser chamados de outra parte do código como se fossem comandos.
- Cada subprograma deve saber como chamar o outro.
- Após seu término, a execução continua a partir do ponto onde foi chamado.

Programação Modular

- Este tipo de programação permite:
- ◆ Organizar o código;
 - ◆ Facilitar o entendimento dos programas;
 - ◆ Facilitar a manutenção ou alterações no código;
 - ◆ Reutilizar o código.

Funções

Conjunto de declarações que são agrupadas para executar uma operação.

```
System.out.println("Algoritmos")
```

Quando chamamos este comando, diversas operações são realizadas até que a mensagem é impressa na tela



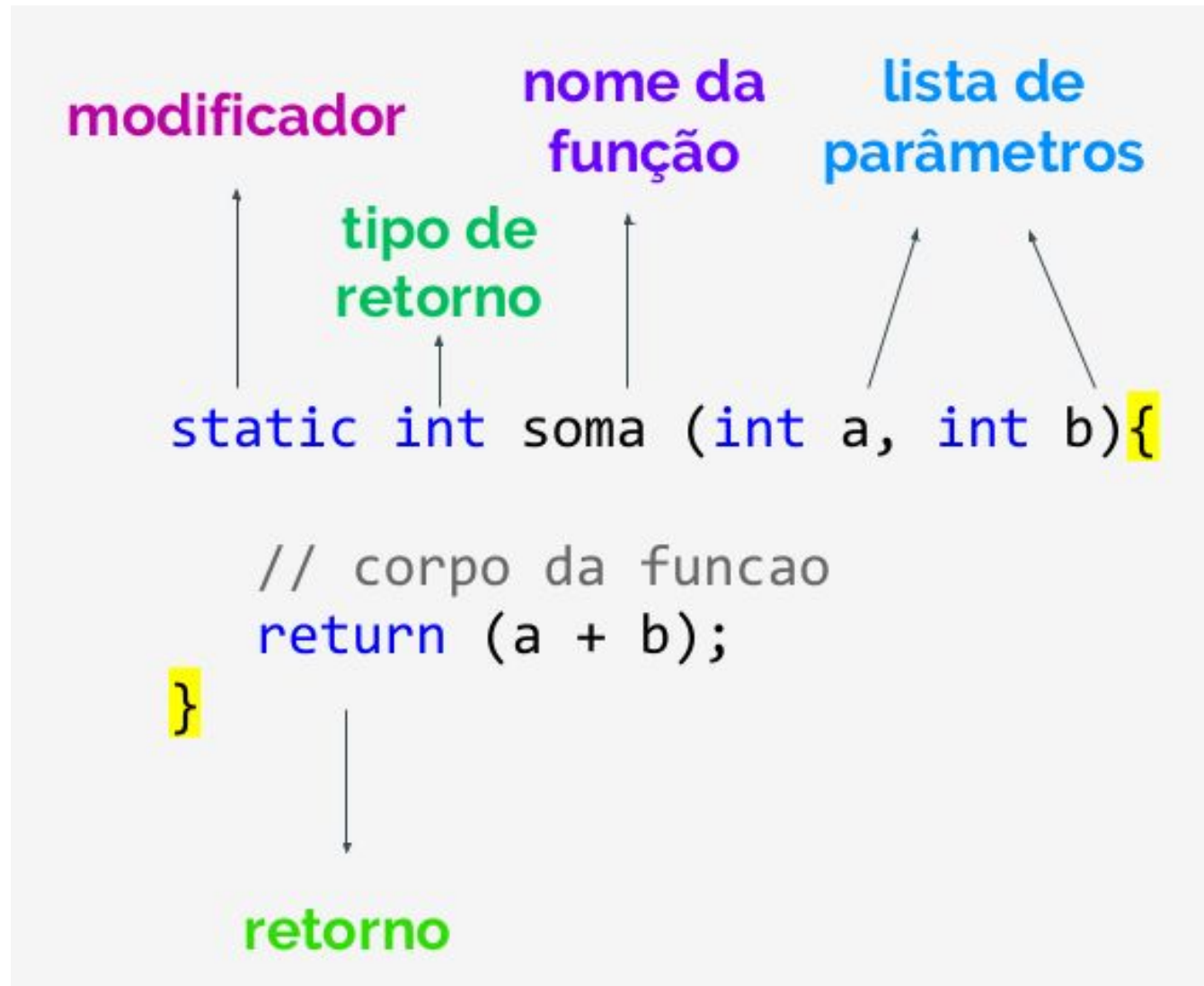
DIVERSAS OPERAÇÕES



Funções

- Uma função é um subprograma que:
 - ◆ Tem um nome;
 - ◆ Define um conjunto de **parâmetros** que precisa receber;
 - ◆ Executa um determinado conjunto de comandos, utilizando estes parâmetros e retorna um único valor.
 - ◆ Define o tipo de dado do valor de retorno.
- Os parâmetros representam a informação que a função requer para efetuar seu trabalho.

Sintaxe de uma Função



Exemplo 01

```
public class ExemploFuncao01 {  
    // declaração da função mínimo  
    static int minimo (int a, int b) {  
        int min;  
        if (a < b) {  
            min = a;  
        }  
        else {  
            min = b;  
        }  
        return min;  
    }  
}
```

Exemplo 01 (cont.)

```
// função main
```

```
public static void main(String[] args) {
```

```
    int x = 11;
```

```
    int y = 6;
```

```
    int m = minimo(x, y);
```

```
    System.out.println("Minimo: " + m);
```

```
}
```

```
}
```

Os valores das variáveis
x e y são copiados para
a função minimo

Tipo void

Permite criar funções sem retorno

```
static void imprime (String x) {  
    System.out.println(x);  
}
```

Exemplo 02

```
public class ExemploFuncao02 {  
    // Apresenta a situação do aluno  
    static void imprimirSituacao (float nota) {  
        if (nota >= 6.0f) {  
            System.out.println("Aprovado");  
        }  
        else {  
            System.out.println("Reprovado");  
        }  
    }  
  
    public static void main(String[] args) {  
        imprimirSituacao(6.7f);  
    }  
}
```

Escopo de Variável

- O escopo de uma variável é o local onde ela pode ser utilizada.
- Uma variável pode ser:
 - ◆ Local: Definida dentro de uma função;
 - ◆ Global: Definida na classe, fora de qualquer função;
 - Evite o uso de variável global!

Exemplo Variável Local

```
public class ExemploVariavelLocal {  
    static void imprimirSituacao (float nota) {  
        String status;  
        if (nota >= 6.0f) {  
            status = "Aprovado";  
        }  
        else {  
            status = "Reprovado";  
        }  
        System.out.println(status);  
    }  
  
    public static void main(String[] args) {  
        imprimirSituacao(6.7f);  
    }  
}
```

status é uma variável local.
Só pode ser usada dentro da
função imprimirSituacao.

Exemplo Variável Local

```
public class ExemploVariavelLocal {  
    static void imprimirSituacao (float nota) {  
        String status;  
        if (nota >= 6.0f) {  
            status = "Aprovado";  
        }  
        else {  
            status = "Reprovado";  
        }  
        System.out.println(status);  
    }  
    public static void main(String[] args)  
    {  
        imprimirSituacao(6.7f);  
        System.out.println(status);  
    }  
}
```

status NÃO pode ser usada aqui. Ela não existe na função main.

Exemplo Variável Global

```
public class ExemploVariavelGlobal {  
  
    static String status;  
  
    static void imprimirSituacao(float nota) {  
        if (nota >= 6.0f) {  
            status = "Aprovado";  
        } else {  
            status = "Reprovado";  
        }  
    }  
  
    public static void main(String[] args) {  
        imprimirSituacao(6.7f);  
        System.out.println(status);  
    }  
}
```

Exercícios

1. Modificar a função `imprimirNota` do exemplo dado para retornar uma `String`. Se aluno aprovado, retornar "Aprovado". Caso contrário, retorna "Reprovado".
2. Criar uma função com dois parâmetros numéricos capaz de verificar qual número é o maior e retorná-lo.
3. Criar uma função com parâmetro capaz de verificar se um número é par ou ímpar.

Exercícios

4. Criar uma função capaz de calcular o resultado para a função matemática abaixo (a função deve receber "x" e retornar a resposta).

$$f(x) = \begin{cases} |x + 2| & , \text{ se } x < -1 \\ x^2 - 4 & , \text{ se } -1 \leq x \leq 2 \\ 2x - 1 & , \text{ se } x > 2 \end{cases}$$

Exercícios

5. Usando funções, construir uma calculadora:

- a. Criar uma função "soma" que recebe dois números e retorna a soma;
- b. Criar uma função "multiplicacao" que recebe dois números e retorna a multiplicação;
- c. Criar uma função "subtracao" que recebe dois número (a e b) e retorna a subtração de a por b;
- d. Criar uma função "divisao" que recebe dois número (a e b) e retorna a divisão de a por b;
- e. Criar uma função "raiz" que recebe um número e retorna a raiz quadrada desse número.
- f. Criar uma função "potencia" que recebe dois números (a e b) e retorna a potência de a por b (a^b)

Exercícios

- g. Por fim, função principal, criar um menu para obter dados do usuário e usar as funções criadas. Exemplo:

Digite:

(1) Soma

(2) Multiplicação

etc.