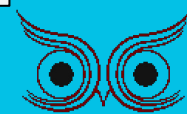




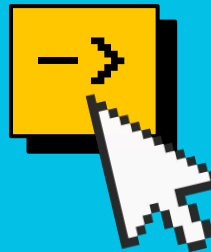
COMPETEC





Aula Extra nº01

Métodos
Funções



Assuntos da aula

- >O que são funções
- >Estrutura de métodos e funções
- >Como utilizar
- >Boas práticas

Ok

Cancel

Options<<

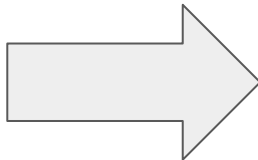


Funções e/ou métodos são blocos de código reutilizáveis que executam uma tarefa específica quando chamados. Eles ajudam a organizar o código, evitar repetição e melhorar a manutenção.

Em linguagens orientadas a objetos como Java, um método é uma função que pertence a uma classe ou objeto. Em outras linguagens, funções podem existir independentemente de classes.

Relação entre Métodos e Funções:

Função
Bloco de código
reutilizável



Método
Função associada a um
objeto/classe

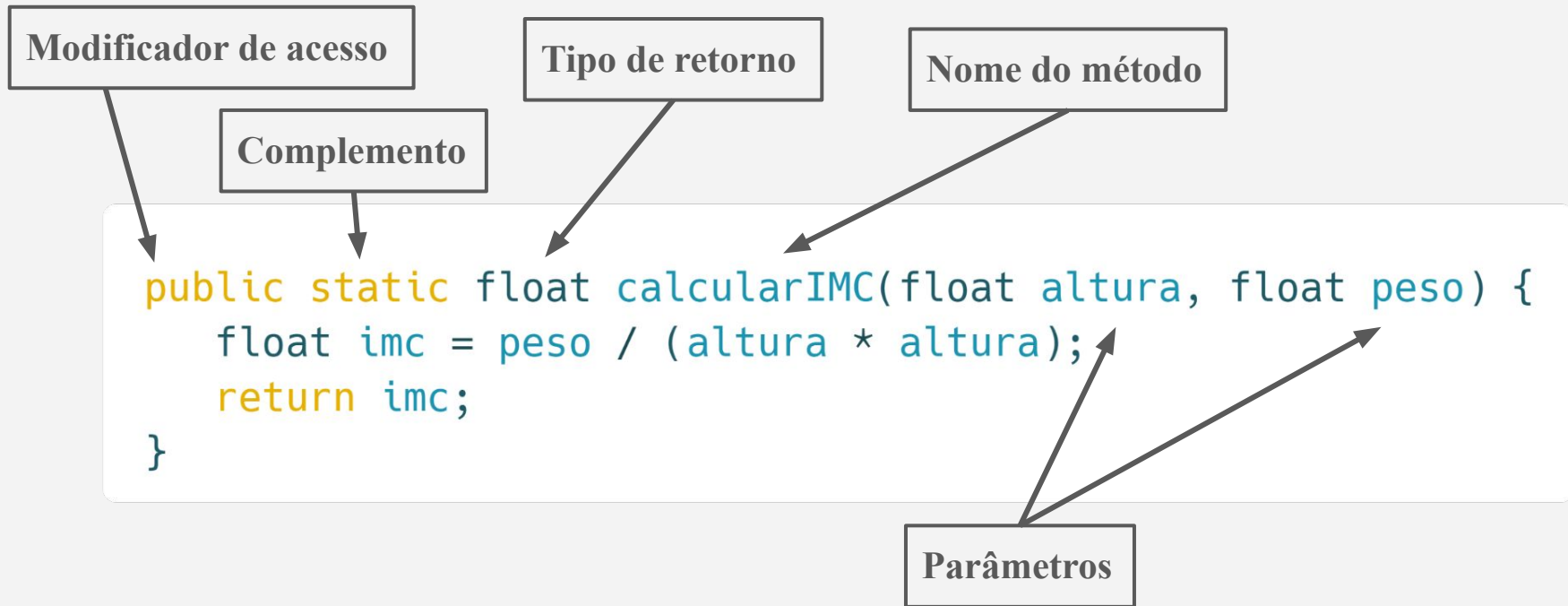
Por que usar Métodos?

- **Reutilização de Código:** Escreva uma vez, use várias vezes. Evita duplicação de código e reduz o tamanho do programa.
- **Organização e Legibilidade:** Divide o código em blocos lógicos e nomeados, tornando-o mais fácil de entender e manter.
- **Manutenção Simplificada:** Alterações precisam ser feitas em apenas um lugar, reduzindo erros e tempo de desenvolvimento.

Cabeçalho do Método



A estrutura do cabeçalho do método é composta por algumas informações sobre ele. É importante se atentar a ordem em que essas informações devem ser colocadas.



Cabeçalho do Método



Componente	Descrição	Exemplos
Modificador de acesso	Define a visibilidade do método	public, private, protected
Complemento	Modificadores adicionais	static, final, abstract
Tipo de retorno	Tipo de dado que o método retorna	void, int, String, boolean
Nome	Identificador do método	calcularMedia, imprimeIMC
Lista de parâmetros	Dados recebidos pelo método	(float altura, float peso)

Para declarar, ou seja, criar um novo método, é importante sempre lembrar de colocar o **cabeçalho** do mesmo, seguido de chaves. Entre as chaves está o que chamamos de *corpo* do método, coloque nele todo o código necessário com os procedimentos do método.

```
public static void imprimeIMC(float altura, float peso) {  
    float imc = peso / (altura * altura);  
    System.out.println("Seu imc eh: " + imc);  
}
```

```
public static boolean _____(float n1, float n2){  
    float m = (n1 + n2) / 2;  
    return (m >= 5);  
}
```

Perguntas:

1. O que esse código faz?
2. Como você o descreveria?
3. Qual nome você daria para esse método?

Basicamente, se o método for void, ele pode ser chamado colocando em uma linha o seu nome e os atributos necessários da função.

```
imprimeIMC(altura, peso);
```

Se o método não for void, seu valor de retorno pode ser atribuído a uma variável.

```
int idade = getIdade(id);
```

Seu valor também pode ser utilizado diretamente em uma operação ou comparação.

```
if (aprovado(nota1, nota2)){  
    System.out.println("Aluno aprovado!");  
}
```

Seguir boas práticas na criação de métodos melhora a qualidade, legibilidade e manutenção do código.

- **Nomes descritivos:** Use nomes que descrevam claramente o que o método faz: `calcularMedia()` em vez de `calc()`
- **Responsabilidade única:** Cada método deve fazer apenas uma coisa, mas fazê-la bem. Separe operações distintas em métodos diferentes.
- **Tamanho adequado:** Mantenha métodos curtos e focados. Se um método ficar muito grande, considere dividi-lo em métodos menores.

COMPETEC



Obrigado pela
atenção! :)

☐

Don't show this message again

Ok

