



## Funções

---

Lógica de Programação I

# Funções

Funções ou Métodos

Em Java, chamamos/damos o nome de **métodos** para funções, que são blocos de códigos responsáveis por executar ações ou procedimentos em seu projeto. Os métodos em Java devem sempre ser definidos em uma classe e é composto por modificador de acesso, tipo do retorno, nome do método e seus argumentos, veja a seguir a sintaxe de um método.

```
[modificador] tipo_retorno nome ([argumentos]) {  
  
    //conteudo do método  
  
}
```

## Modificadores de Acesso

São utilizados para definir o tipo de acesso que aquele método terá. Podem ser:

- **public:** Pode ser acessado por qualquer outra classe do projeto.
- **protected:** Podem ser acessados pelos métodos da própria classe, pelas classes do mesmo pacote e pelas classes derivadas.
- **private:** Só podem ser acessados pelos métodos da própria classe.

## Modificadores de Acesso

Ainda existe uma quarta opção que é quando omitimos o modificador. Nesse caso, chamamos de visibilidade "*default*", a qual permite o acesso pelos métodos da própria classe e do mesmo pacote.



# Modificadores de Acesso

Exemplos:

```
public int somaDoisNumeros (int numero1, int numero2){  
    return numero1 + numero2;  
}  
  
private int subtrairDoisNumeros (int numero1, int numero2){  
    return numero1 - numero2;  
}  
  
protected int dividirDoisNumeros (int numero1, int numero2){  
    return numero1 / numero2;  
}
```

## Nome do Método

É o nome que damos para aquele bloco de código, o nome que damos a um método deve remeter a aquilo que o método realiza, se você cria um método que tem como objetivo somar dois números, você pode colocar o nome de `somaDoisNumeros`.

## Tipo de retorno

É o tipo que será retornado ao final da execução daquele método para quem o chamar, como `int`, `string`, utilizamos `void`, quando o retorno será vazio.

## Argumentos

Também chamados de parâmetros, são valores que devem ser passados ao método quando chamados, um método pode ter nenhum, um ou mais parâmetros. O nome do método acrescido de seus parâmetros é denominado **assinatura do método**.



## Quando utilizar métodos?

Podemos utilizar métodos quando queremos isolar alguma execução ou funcionalidade em um projeto, principalmente se for algo que precisará ser executado muitas vezes e/ou em diferentes partes do código, com isso não escrevemos código repetido e podemos apenas chamar o método criado onde queremos executar essa ação. Segue exemplo de um método.

```
public int somaDoisNumeros (int numero1, int numero2) {  
    return numero1 + numero2;  
}
```

## Método main

Quando um programa Java é executado, a JVM tenta sempre chama o método `main` da classe onde geralmente iniciamos o nosso projeto.

Quando declaramos a `main` como `static` é permitido que a JVM chame a classe `main` sem precisar criar uma instância da mesma, dessa forma, ela é conhecida como classe principal do projeto.

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

# Exercícios

## Exercícios

**Exercício 1:** Qual modificador de acesso permite que um método seja acessado por qualquer classe em qualquer pacote?

- a) `private`
- b) `public`
- c) `protected`
- d) Nenhum modificador (sem especificação)



**Exercício 2:** Em relação à visibilidade, qual modificador de acesso restringe o acesso a um método apenas à classe onde ele está definido?

- a) `private`
- b) `protected`
- c) `public`
- d) Modificador padrão (default)

**Exercício 3:** Qual é o modificador de acesso que permite que um método seja acessível a classes no mesmo pacote e subclasses em qualquer pacote?

- a) public
- b) private
- c) protected
- d) Modificador padrão (default)

**Exercício 4:** Quando nenhum modificador de acesso é definido para um método em Java, qual modificador de acesso é aplicado implicitamente?

- a) public
- b) private
- c) protected
- d) Modificador padrão (default)

### **Exercício 5: Função com Enum e Scanner**

Crie um enum chamado **MesAno** representando os meses do ano.

Escreva uma função que recebe um mês como entrada e ano, utilize um Scanner para obter um ano e o mês escolhido.

A função deve imprimir a quantidade de dias no mês e considerar se o ano é bissexto.



Obrigada