

# Métodos em Java



Prof. Dr. João Paulo Lemos Escola  
Copyright© 2022

# Tópicos da aula

- Métodos;
- Parâmetros e argumentos;
- Sobrecarga.

# Introdução

- Métodos são como as funções em linguagens estruturadas;
- Servem para tornar mais prático o código, que pode ser executado diversas vezes por meio da declaração de um trecho repetitivo;
- Métodos que não retornam valor devem ser definidos como **void** e os que retornam valor devem ser declarados com o tipo de dado de retorno, que pode ser primitivo ou uma Classe;

# Formato de um método

- Métodos tem o seguinte formato:

```
qualificador tipo-de-dado nome([lista de parâmetros]){  
    // código a executar  
}
```

- Exemplo de método sem retorno e sem parâmetros:

```
public static void latir(){  
    System.out.println("Au!");  
}
```

- Como usar o método:

```
latir();
```

# A06ex01.jar

- Crie um programa que contenha o método latir();
- Adicione ao programa o método miar() e o método mugir()

# Métodos com parâmetros

- Exemplo de método sem retorno e que recebe parâmetros:

```
public static void latir(boolean muito){  
    if (muito)  
        System.out.println("Au, au, au, au...");  
    else  
        System.out.println("Au!");  
}
```

- Como usar o método:

```
latir(true); // latir muito  
latir(false); // latir pouco
```

# Métodos com parâmetros (cont.)

- Exemplo de método sem retorno e que recebe mais de um parâmetro:

```
public static void latir(boolean muito, String latido){  
    if (muito)  
        System.out.println(latido+", "+latido+", "+latido+",  
        "+latido+"...");  
    else  
        System.out.println(latido+"!");  
}
```

# A06ex02.jar

- Crie um programa que utilize o método latir com o parâmetro booleano;
- Altere o programa para utilizar o método latir com dois parâmetros.



# Métodos com retorno

- Exemplo de método com retorno:

```
public static String getResultado(float nota){  
    if (nota > 5)  
        return "Aprovado";  
    else  
        return "Reprovado";  
}
```

# Métodos com retorno (cont.)

- Outro exemplo de método com retorno:

```
public static String getResultado(float nota1, float nota2){  
    String resultado;  
    float media = (nota1+nota2)/2;  
    if (media > 5)  
        resultado = "Aprovado";  
    else  
        resultado = "Reprovado";  
  
    return resultado;  
}
```

# A06ex03.jar

- Crie um programa que implemente o método getResultado no primeiro formato;
- Comente o método (desabilitando-o) e crie um novo que implemente o recebimento de dois parâmetros.

# Sobrecarga

- Quando temos mais de um método, com o mesmo nome e quantidades/tipos diferentes de parâmetros:

```
public static void latir(){  
    System.out.println("Au!");  
}
```

```
public static void latir(boolean muito){  
    if (muito)  
        System.out.println("Au, au, au, au...");  
    else  
        System.out.println("Au!");  
}
```

# A06ex04.jar

- Implemente sobrecarga do método latir, conforme exposto no slide anterior.

# Acesso a métodos de outras classes

- Podemos acessar métodos de outras classes utilizando o formato abaixo:

```
nome-da-classe.nome-do-metodo
```

- Não é necessário, ainda, instanciar objeto da classe se o método tiver o qualificador “static”.
- Sem o qualificador “static” será necessário instanciar um objeto e executar o método a partir dele:

```
Cachorro bilu = new Cachorro("Preto");  
bilu.latir();
```

# Classe Cachorro

```
public class Cachorro{  
    public void latir(){  
        System.out.println("Au!");  
    }  
  
    public void latir(boolean muito){  
        if (muito)  
            System.out.println("Au, au, au, au...");  
        else  
            System.out.println("Au!");  
    }  
}
```

# A06ex05

- Crie a classe “Cachorro” implementando a sobrecarga de métodos;
- Crie outra classe, chamada de Gato, que possua sobrecarga no método “miar”.



# Métodos com Array

- Podemos criar métodos que utilizam array como parâmetro ou como retorno:

```
public static String[] latir(boolean muito){  
    if (muito)  
        return {"Au, au, au, au..."};  
    else  
        return {"Au!"};  
}
```

- Utilizando o método:

```
String[] latidos = latir(true);  
for (String latido: latidos)  
    System.out.println(latido);
```

# A06ex06.jar

- Crie um programa que receba como parâmetro um array de nomes de times e retorne o mesmo array ordenado e com os nomes em letras maiúsculas.

# A06ex07.jar

- Crie um método getProdutos que receba o código fictício do fabricante (1,2 ou 3) e retorne um ArrayList de Produtos (String).

# O que aprendemos?

- Métodos;
- Parâmetros e argumentos;
- Sobrecarga.

# Na próxima aula...

- Objetos;
- Classes;
- Escopo;
- Encapsulamento;
- Construtor;
- Classe como atributo.