



Sobrecarga e sobreescrita



**Certified
Developer**
The Ultimate Tech Degree

DigitalHouse >
Coding School



Temas

1

Classes

2

Sobrecarga

3

Sobreescrita



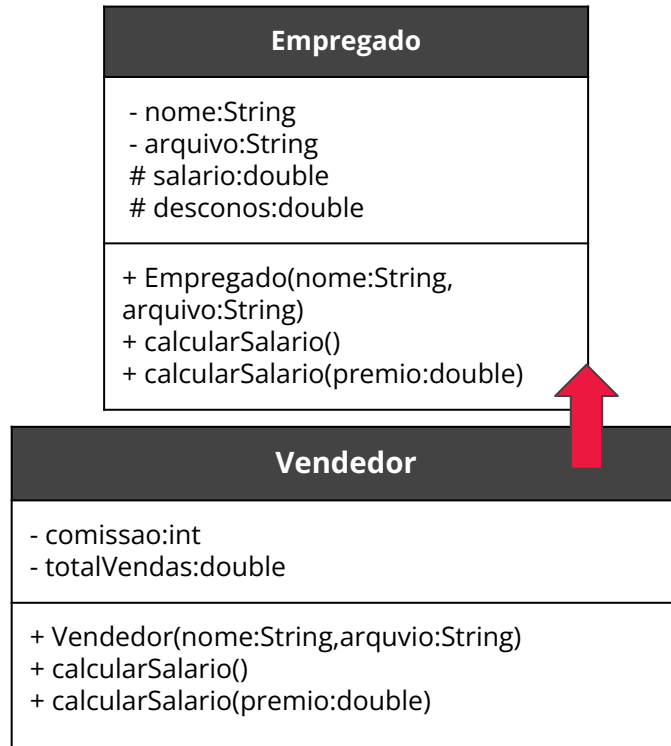
1 | Classes



Classes

Um empregado tem um arquivo e um salário. Seu salário pode ser calculado simplesmente subtraindo os descontos do salário base. Ocasionalmente, ele recebe um prêmio, então o cálculo do salário nesse caso varia, pois esse prêmio teria que ser adicionado.

Por outro lado, existem vendedores, que também são empregados, neste caso, além do salário-base, recebem uma comissão por venda.



2

| Sobrecarga



Mesmo nome comportamento diferente

Lembre-se de que só podemos sobrecarregar um método se sua assinatura mudar.
Como vemos neste caso:

```
public class Empregado{  
    private String nome;  
    private String arquivo;  
    protected double salario;  
    protected double descontos;  
  
    public double calcularSalario(){  
        return salario - descontos;  
    }  
    public double calcularSalario(double premio){  
        return salario - descontos + premio;  
    }  
}
```



Quando cada um é usado

Ao usar os métodos, utilizam-se os parâmetros que são passados como base, ou seja, o método cuja assinatura corresponde aos parâmetros usados será chamado.

```
public class Main {  
  
    public static void main(String[] args) {  
  
        Empregado meuEmpregado = new Empregado("Jose", "1111");  
        System.out.println("Salario a pagar: " + meuEmpregado.calcularSalario());  
        System.out.println("Salario a pagar: " + meuEmpregado.calcularSalario(5000));  
  
    }  
}
```

3

Sobreescrita



Classe Vendedor

A classe Vendedor, com os métodos sobrescritos, é necessário dar outro comportamento para um Vendedor. Portanto, surge a necessidade de sobrescrevê-los.

```
public class Vendedor extends Empregado{
    private int comissao;
    private double totalVendas;

    @Override
    public double calcularSalario(){
        return salario-descontos + totalVendas/100*comissao;
    }

    @Override
    public double calcularSalario(double premio){
        return salario-descontos + premio + totalVendas/100*comissao;
    }
}
```



Sobrescrita

```
@Override
public double calcularSalario(){
    return salario-descontos + totalVendas/100*comissao;
}

@Override
public double calcularSalario(double premio){
    return salario-descontos + premio + totalVendas/100*comissao;
}
```

@Override nos diz que o comportamento anterior do método foi sobrescrito, e estamos sobrescrevendo para dar a ele uma maneira diferente de resolver. Para objetos Vendedor, o método a ser executado é este, que sobrescreve o anterior.

DigitalHouse>
Coding School