



## Aula 8

# Classe Abstrata e Exercícios

Iniciativa Conjunta:



Com o apoio de:





# O que é uma classe abstrata?

- As classes abstratas servem como “modelo” para outras classes que dela herdem, não podendo ser instanciadas por si só.
- Uma classe abstrata pode ter métodos abstratos e não abstratos. Os métodos abstratos são métodos sem implementação, que terão de ser implementados nas subclasses.
- Resumindo: classes abstratas não podem ser instanciadas, elas servem apenas para que outras classes a usem como modelo (a estendam e herdem os atributos/propriedades e métodos delas).

# O que é um método abstrato?

```
public abstract class ContaBancaria {  
  
    ...  
  
    public abstract double getSaldo();  
}  
}
```

Apenas é definida a assinatura. A implementação só é feita pelas subclasses

- Exemplo: classe abstrata para representar uma conta (*Account*), com um método abstrato *getSaldo()* que deve ser implementado pelas suas subclasses.

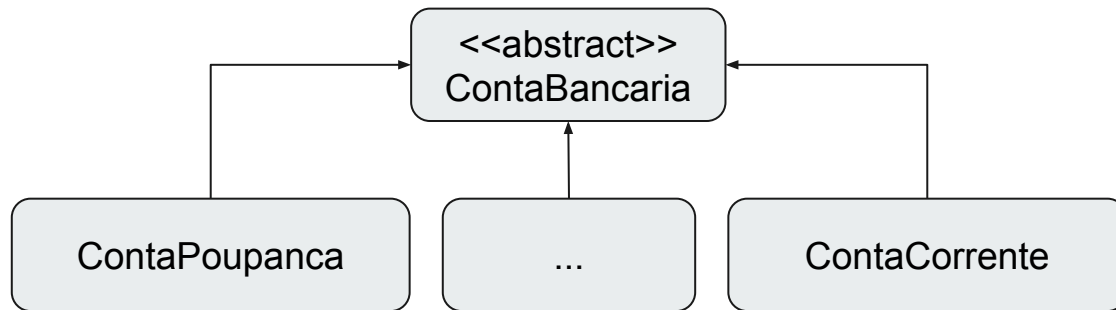


# Exemplo

```
public abstract class ContaBancaria {  
  
    ...  
  
    public abstract double getSaldo();  
  
}
```

```
public class ContaPoupanca extends ContaBancaria {  
  
    ...  
  
    @Override  
    public double getSaldo(){  
  
        return saldo * juro;  
  
    }  
  
}
```

# Exemplo



# Exemplo

```
Account acc = new Account(); }
```

Não é possível instanciar  
uma classe abstrata!

```
Account acc = new ContaPoupanca(); }
```

```
ContaPoupanca acc = new ContaPoupanca();
```

Está a ser guardado com  
o tipo **Account** ->  
**polimorfismo**. Só é  
possível aceder aos  
métodos definidos na  
classe mãe



# Exercício 1

Crie uma hierarquia de classes para representar os diferentes tipos de funcionários de um escritório que tem os seguintes cargos: **gerente**, **assistente** e **vendedor**. Escreva uma classe base abstrata chamada **Funcionário** que declara um método abstrato:

```
double calculaSalario()
```

Esta classe também deve definir os seguintes atributos: nome (tipo String), código de funcionário (tipo String) e salário base (tipo double). Crie um construtor que recebe os valores a serem armazenados nos respectivos atributos. Esta classe abstrata deverá ser estendida pelas outras classes representativas dos tipos de funcionários. Em cada classe deve-se sobrescrever o método `calculaSalario` de forma que o cálculo do salário seja feito da seguinte forma: o gerente recebe duas vezes o `salário_base`, o assistente recebe o `salário_base` e o vendedor recebe o `salário_base` mais uma comissão definida no momento de inicialização.

Crie uma classe *Teste* com um método `main` que cria um objeto de cada tipo e imprime para o ecrã os salários de cada um.



## Exercício 2

Escreva uma classe abstrata chamada *CartaoPresente*. Essa classe representa todos os tipos de cartões presente e conterá apenas um atributo: *destinatario* (tipo `String`).

Nessa classe, deverá também ser declarado o método *public abstract void showMessage()*.

Crie classes filhas da classe *CartaoPresente*: *DiaDosNamorados*, *Natal* e *Aniversário*. Cada uma dessas classes deve conter um método construtor que receba o nome do destinatário do cartão. Cada classe também deverá implementar o método *showMessage()*, mostrando uma mensagem ao usuário com seu nome e que seja específica para a data de comemorativa do cartão.

Crie uma classe *Teste* com um método *main* onde são criadas instâncias dos 3 tipos de cartões. Depois, exiba as mensagens desse cartão chamando o método *showMessage()*.