

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
INSTITUTO METRÓPOLE DIGITAL

AULA 06

LINGUAGEM DE PROGRAMAÇÃO 2
JAVA



PROF. JANIHERYSON FELIPE

CONTEÚDO DESSA AULA

- **ENTENDER O CONCEITO DE ABSTRAÇÃO;**
- **APLICAR OS CONCEITOS DE ABSTRAÇÃO EM CLASSES E METODOS;**
- **CONHECER OS CONCEITOS DE INTERFACES NO JAVA;**
- **DISCUSSÕES E DÚVIDAS GERAIS.**

CLASSES E METODOS ABSTRATOS EM JAVA

- Abstração é um dos **pilares fundamentais da programação orientada a objetos (POO)**. Em Java, a abstração é alcançada através de classes e métodos abstratos.



CLASSES E METODOS ABSTRATOS EM JAVA

Classes abstratas:

- Uma classe abstrata é uma classe que não pode ser **instanciada diretamente**, ou seja, você **não pode criar objetos diretamente** a partir dela. Ela serve como um modelo para outras classes derivadas.
- Uma classe abstrata pode conter **métodos abstratos** (métodos sem implementação) e **métodos concretos** (com implementação).

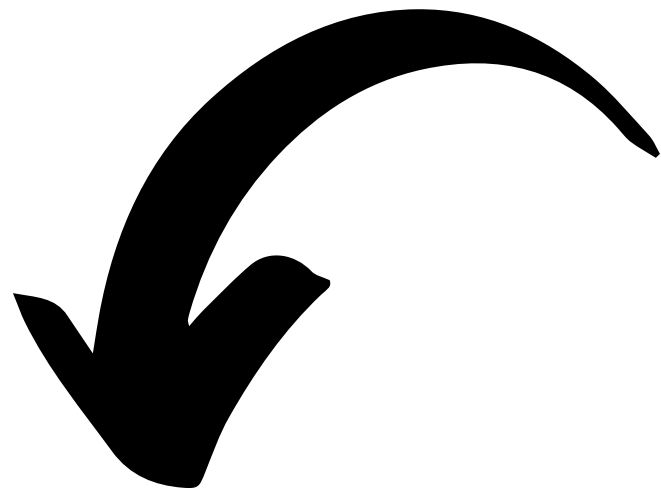
CLASSES E METODOS ABSTRATOS EM JAVA

Métodos abstratos:

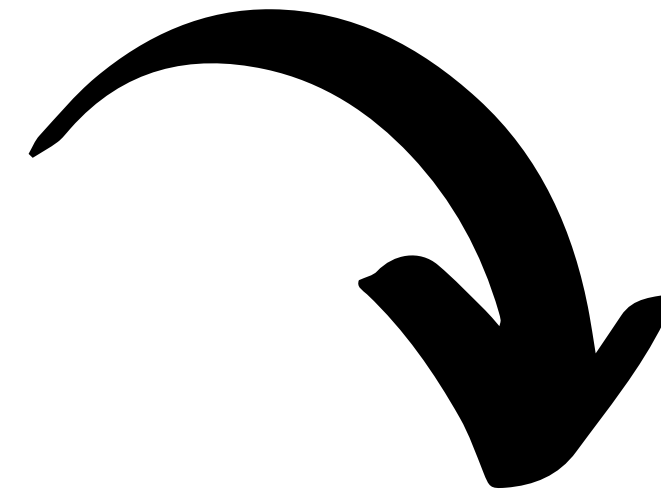
- Um **método abstrato** é um método que **não possui implementação** na classe onde é declarado. Ele é declarado utilizando a palavra-chave **abstract** e não possui corpo. As subclasses que estendem uma classe abstrata devem fornecer implementações para todos os métodos abstratos da classe pai.



CLASSES E METODOS ABSTRATOS EM JAVA



```
abstract class Forma {  
  
    abstract double calcularArea();  
  
    abstract double calculaPerimetro();  
  
    void outroMetodo(){  
        System.out.println("Faz algo...");  
    }  
}
```



```
class Retangulo extends Forma {  
    double base;  
    double altura;  
  
    @Override  
    double calcularArea() {  
        return base * altura;  
    }  
  
    @Override  
    double calculaPerimetro() {  
        return base * 2 + altura * 2;  
    }  
}
```

```
class Circle extends Forma {  
    double raio;  
  
    double calcularArea() {  
        return Math.pow(raio, 2) * Math.PI;  
    }  
  
    @Override  
    double calculaPerimetro() {  
        return 2 * Math.PI * raio;  
    }  
}
```


INTERFACES EM JAVA

- Uma interface é uma **coleção de métodos abstratos** que definem um **contrato** para uma classe que a **implementa**. Ela fornece um meio de especificar um comportamento que uma classe deve implementar sem definir como esse comportamento será realizado.
- Uma interface é declarada usando a palavra-chave “**interface**”. Para implementar uma interface, uma classe usa a palavra-chave **implements**. Uma classe **pode implementar várias interfaces**.

INTERFACES EM JAVA

```
public interface MinhaInterface {  
  
    void metodo1();  
    void metodo2();  
  
    int VALOR = 10;  
  
}
```

```
class MinhaClasse implements MinhaInterface {  
    public void metodo1() {  
    }  
  
    public void metodo2() {  
  
    }  
  
}
```



INTERFACES EM JAVA

- **Métodos Abstratos:** Uma interface pode conter **apenas métodos abstratos** (sem implementação). Uma classe que implementa uma interface deve fornecer uma implementação para todos os métodos abstratos na interface.
- **Constantes:** As interfaces podem conter constantes que são implicitamente públicas, estáticas e finais. Estas constantes podem ser acessadas usando o nome da interface. Por exemplo, `MinhaInterface.CONSTANTE`.

INTERFACES EM JAVA

- **Múltipla Herança:** Java não permite herança múltipla de classes, mas uma classe pode implementar múltiplas interfaces. Isso permite que as classes possuam comportamentos de várias fontes.
- **Extensão de Interfaces:** Uma interface pode estender outras interfaces usando a palavra-chave **extends**. Isso permite que uma interface herde métodos de outras interfaces. Por exemplo:

