



# Variáveis e Métodos da Classe



**Certified  
Developer**  
The Ultimate Tech Degree

**DigitalHouse** >  
Coding School



# Temas

**1**

**Classes e Objetos**

**2**

**Variáveis da Classe**

**3**

**Métodos da Classe**

**4**

**Exemplos**



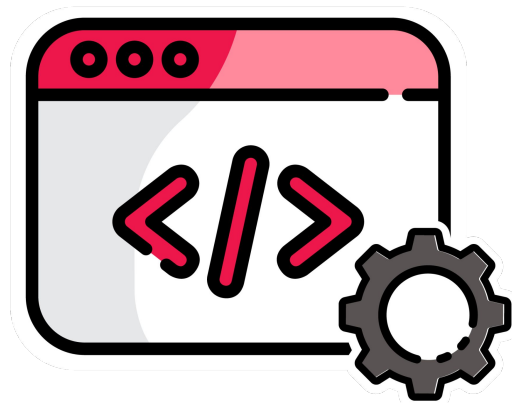
# 1 | Classes e Objetos



# Classes

Vimos que a abordagem de programação orientada a objetos é baseada na identificação de objetos com seus atributos e responsabilidades.

Assim, constatamos que existem grupos de objetos que, embora tenham estados diferentes - valores de atributos -, têm em comum o que são os atributos e quais são as suas responsabilidades. Portanto, esse "**molde**" é o que chamamos de **classes**.





# Objetos



## Engenheiro

- nome: String
- matricula: int
- salario: float

- + construir()
- + cobrar()

Classe Engenheiro

João Milutti  
Mat. 434244  
12.000 R\$/mes

Miguel Zapata  
Mat. 1133242  
13.000 R\$/mes

Claudia Vasques  
Mat. 775543  
14.000 R\$/mes

Muriel Fringuesti  
Mat. 242352  
16.000 R\$/mes

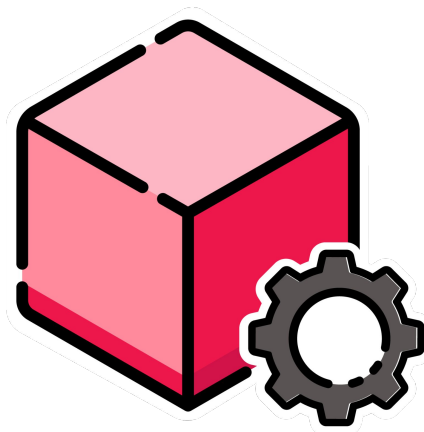
Instâncias ou objetos Engenheiro



# O que os objetos compartilham?

Todos **os objetos** de uma classe possuem **a mesma estrutura**: os mesmos **atributos** e o mesmo **comportamento**, ou seja, podem fazer a mesma coisa.

Cada objeto tem seus próprios atributos, pode ter diferentes valores em seus atributos, tem seu próprio estado.



## **2 | Variáveis da classe**



# Um valor comum para todos os objetos

Todos os objetos podem ter um valor comum?



Todos os diferentes caminhões usam o mesmo valor de combustível.



Todos os corredores de 100 metros tentam quebrar o mesmo recorde mundial.





# Variáveis da classe

Vamos chamar **variáveis de classe** para essas variáveis — atributos — que armazenam valores comuns a todos os objetos. O combustível é o mesmo para qualquer objeto Caminhão.

Caminhao
<ul style="list-style-type: none"><li>- marca: String</li><li>- patente: String</li><li>- <u>valorCombustivel: double</u></li></ul>
<ul style="list-style-type: none"><li>+ Caminhao (String marca, String patente)</li><li>+ transportar()</li><li>+ gastoCombustivel(int litros)</li></ul>

O nome da variável é **sublinhado** para indicar que é uma variável de classe.

**3**

## **Métodos de classe**

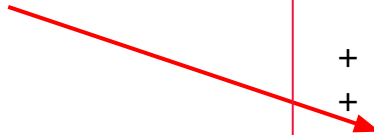


# Métodos de classe

Um método de classe pode ser usado, sem a necessidade de instanciar ou criar um objeto, diretamente com a classe.

Definimos

`mudarPrecoCombustivel(double preco)`,  
como um método de classe. Para indicar  
que é um método de classe, devemos  
também sublinhá-lo.



## Caminhao

- marca: String
  - patente: String
  - valorCombustivel: double
- 
- + Caminhao (String marca, String patente)
  - + transportar()
  - + gastoCombustivel(int litros)
  - + mudarPrecoCombustivel(double preco)

# 4 | Exemplo de uso



# Classe Caminhão

```
public class Caminhao {  
    private String marca;  
    private String patente;  
    static private double valorCombustivel;  
  
    public Caminhao(String marca, String patente){  
        this.marca = marca;  
        this.patente = patente;  
    }  
    public double gastoCombustivel(int litros){  
        return litros * Caminhao.valorCombustivel;  
    }  
    static public void mudarPrecoCombustivel(double preco){  
        Caminhao.valorCombustivel = preco;  
    }  
}
```



# Código

```
public class Caminhao {  
    private String marca;  
    private String patente;  
    static private double valorCombustivel;  
  
    public Caminhao(String marca, String patente){  
        this.marca=marca;  
        this.patente=patente;  
    }  
    public double gastoCombustivel(int litros){  
        return litros*Caminhao.valorCombustivel;  
    }  
    static public void mudarPrecoCombustivel(double preco){  
        Caminhao.valorCombustivel=preco;  
    }  
}
```

Definimos a variável como estática, o que a torna inutilizável com um objeto.



# Código

```
public class Caminhao {  
    private String marca;  
    private String patente;  
    static private double valorCombustivel;  
  
    public Caminhao(String marca, String patente){  
        this.marca=marca;  
        this.patente=patente;  
    }  
    public double gastoCombustivel(int litros){  
        return litros*Caminhao.valorCombustivel;  
    }  
    static public void mudarPrecoCombustivel(double preco){  
        Caminhao.valorCombustivel=preco;  
    }  
}
```

Definimos o método como estático, o que o torna inutilizável com um objeto.



# Código

```
public class Caminhao {  
    private String marca;  
    private String patente;  
    static private double valorCombustivel;  
  
    public Caminhao(String marca, String patente){  
        this.marca=marca;  
        this.patente=patente;  
    }  
    public double gastoCombustivel(int litros){  
        return litros*Caminhao.valorCombustivel;  
    }  
    static public void mudarPrecoCombustivel(double preco){  
        Caminhao.valorCombustivel=preco;  
    }  
}
```

Acessamos a variável de classe e podemos alterar seu valor.





# Programa principal

```
public class Main {  
  
    public static void main(String[] args) {  
  
        Caminhao meuCaminhao = new Caminhao("Ford","AB XXX CD");  
  
        Caminhao.mudarPrecoCombustivel(98.50);  
  
        Sytem.out.println("Gasto " + meuCaminhao.gastoCombustivel(40));  
  
    }  
}
```



# Código

```
public class Main {
```

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

```
        Caminhao meuCaminhao = new Caminhao("Ford", "AB XXX CD");
```

Criamos um objeto da classe Caminhao.

```
        Caminhao.mudarPrecoCombustivel(98.50);
```

```
        System.out.println("Gasto " + meuCaminhao.gastoCombustivel(40));
```

```
    }
```

```
}
```



# Código

```
public class Main {  
  
    public static void main(String[] args) {  
  
        Caminhao meuCaminhao = new Caminhao("Ford", "AB XXX CD");  
  
        Caminhao.mudarPrecoCombustivel(98.50);  
  
        Sytem.out.println("Gasto " + meuCaminhao.gastoCombustivel(40));  
  
    }  
}
```

Usamos o método de classe por meio da classe e não do objeto.

DigitalHouse>  
Coding School