

O QUE É UMA CLASSE?

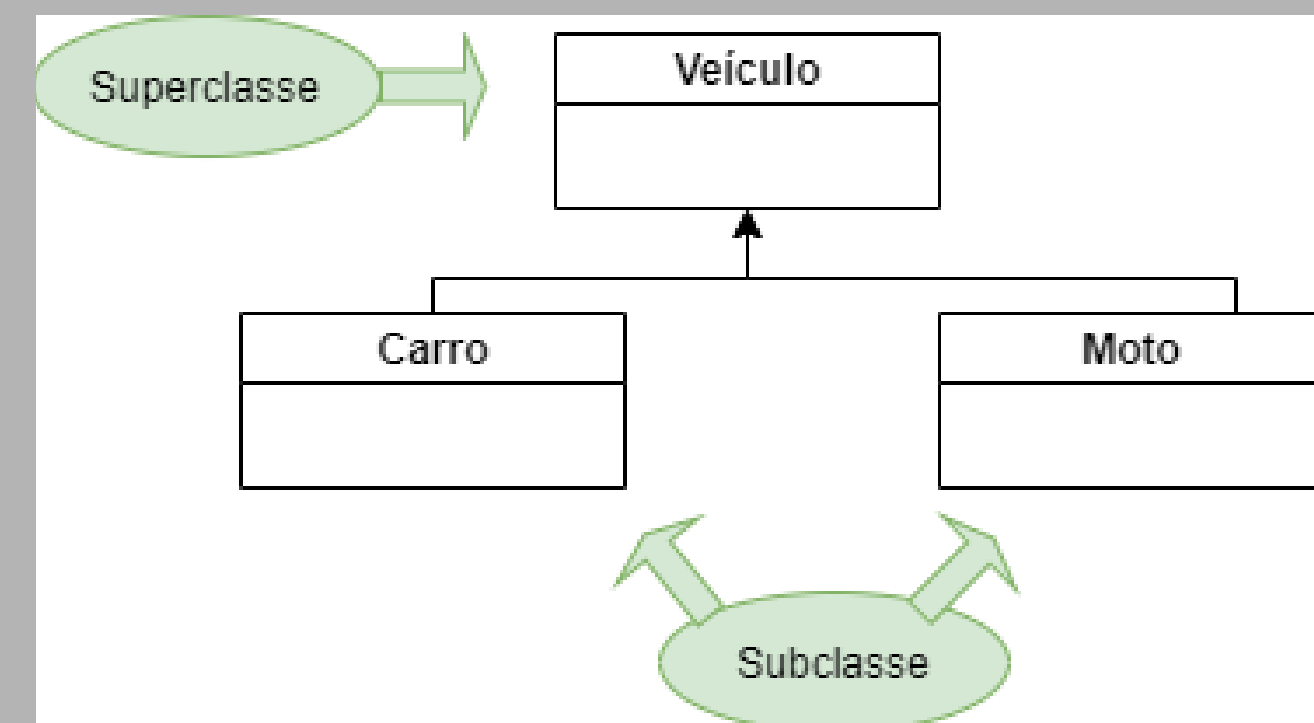
Uma estrutura que define um tipo de objeto. Ela é uma espécie de modelo ou planta a partir da qual podem ser criados objetos reais, que são instâncias da classe.

EX:

```
public class Pessoa {  
    private String nome;  
    private int idade;  
  
    public Pessoa(String nome, int idade) {  
        this.nome = nome;  
        this.idade = idade;  
    }  
  
    public String getNome() {  
        return nome;  
    }  
  
    public int getIdade() {  
        return idade;  
    }  
}
```

O QUE É UMA SUBCLASSE?

uma subclasse é uma classe que herda atributos e métodos de outra classe, chamada superclasse. A subclasse especializa a superclasse, adicionando seus próprios atributos e métodos ou modificando os herdados, enquanto mantém a relação de "é um tipo de" com a superclasse.



O QUE É HERANÇA?

Herança é quando uma classe derivada pega atributos e métodos de uma classe base na programação orientada a objetos. Isso ajuda a reutilizar código e criar relações hierárquicas entre classes.

O QUE SÃO ATRIBUTOS?

Atributos são características ou propriedades que descrevem um objeto. Eles armazenam dados sobre o objeto, definindo seu estado. Pense neles como as variáveis de uma classe que guardam informações específicas sobre cada instância (objeto) criada a partir dessa classe.

O QUE SÃO MÉTODOS?

Métodos são funções associadas a um objeto que definem suas ações ou comportamentos. Eles especificam o que um objeto pode fazer e como ele interage com outros objetos.



EXEMPLO

```
// Classe Animal
public class Animal {
    // Atributos
    String nome;
    int idade;
```

Essa classe
Animal
tem dois atributos
nome(uma String para o nome do animal)
e idade
(um inteiro para a idade). Ela serve para criar objetos que representam animais com essas informações.

```
// Método para apresentar o animal
```

```
public void apresentar() {
    System.out.println("Olá, eu sou um animal chamado " + nome + " e tenho " + idade + " anos.");
}
```

```
// Método que pode ser sobrescrito pelas subclasses
```

```
public void emitirSom() {
    System.out.println("O animal faz um som.");
}
```

```
// Subclasse Cachorro que herda de Animal
```

```
public class Cachorro extends Animal {
    // Método específico do Cachorro
    @Override
    public void emitirSom() {
        System.out.println("Au Au!");
    }
}
```

EXEMPLO

```
// Classe do Personagem 1
class Personagem1 {
    String nome = "Arthas";
    int vida = 100;

    public void gritoDeGuerra() {
        System.out.println(nome + " grita: \"Pelo rei e pela glória!\");
    }

    public void atacar(Personagem2 alvo) {
        gritoDeGuerra();
        int dano = 30;
        System.out.println(nome + " ataca " + alvo.nome + " causando " + dano + " de dano!");
        alvo.vida -= dano;
        if (alvo.vida < 0) alvo.vida = 0;
        System.out.println(alvo.nome + " agora tem " + alvo.vida + " de vida.");
    }
}

// Classe do Personagem 2
class Personagem2 {
    String nome = "Sylvanas";
    int vida = 90;

    public void gritoDeGuerra() {
        System.out.println(nome + " grita: \"A escuridão me fortalece!\");
    }
}

// Classe principal para executar a batalha
public class Batalha {
    public static void main(String[] args) {
        Personagem1 p1 = new Personagem1();
        Personagem2 p2 = new Personagem2();

        // Apenas o Personagem1 ataca o Personagem2
        p1.atacar(p2);
    }
}
```