

JAVA - Projeto Biblioteca

1. Hello World em Java

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Olá, Mundo!");  
    }  
}
```

Em Java, uma **classe** é um modelo ou um projeto que descreve as propriedades e comportamentos de um objeto. As propriedades são representadas por variáveis e os comportamentos são representados por métodos.

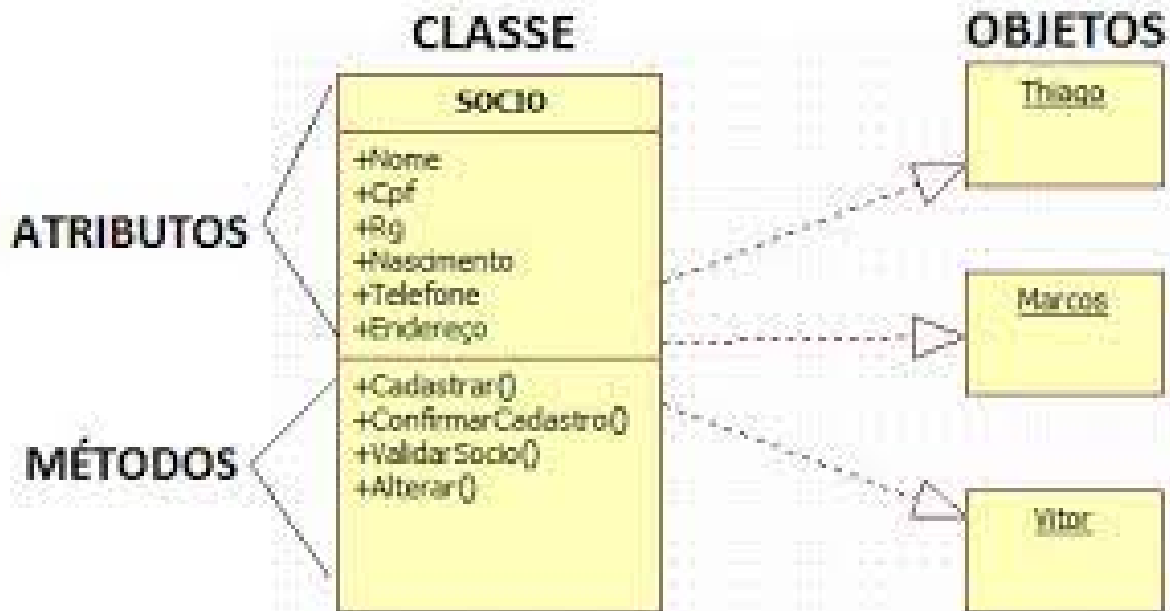
Um **objeto** é uma instância de uma classe. Ele tem um estado e comportamento definidos pela classe. Você pode criar quantos objetos quiser a partir de uma única classe.

Aqui estão alguns exercícios que você pode tentar:

1. **Classe Números:** Esta classe representa um conjunto de números, onde você pode localizar o valor do maior elemento, menor elemento e calcular a média dos elementos do vetor¹.
2. **Classe Conta Bancária:** Esta classe representa uma conta bancária que possui o número da conta e saldo. Ela também deve executar os seguintes métodos: extrato (mostra na tela o número e o saldo da conta), saque (recebe como parâmetro um valor e retira este valor do saldo da conta), depósito (recebe como parâmetro um valor e adiciona este valor ao saldo da conta)¹.
3. **Classe Calculadora:** Esta calculadora deve ter os seguintes métodos: soma (recebe dois números e mostra o valor da soma), subtração (recebe dois números e mostra o valor da subtração entre eles), divisão (recebe dois números e mostra o valor da divisão entre eles), multiplicação (recebe dois números e mostra o valor da

multiplicação entre eles), resto (recebe dois números e mostra o valor do resto da divisão entre esses dois números)1.

4. **Classe Televisor:** Essa classe deve possuir três atributos: canal (inicia em 1 e vai até 16), volume (inicia em 0 e vai até 10), ligado (inicia em desligado ou false). Ela também deve ter os seguintes métodos: aumentarVolume(), reduzirVolume(), subirCanal(), descerCanal(), ligarTelevisor(), desligarTelevisor(), mostraStatus()1.
5. **Classe Animal:** Crie um programa com a classes animal, onde um cachorro, um gato e um passarinho herdam as características da classe animal2.



```
// Criando a classe Carro
public class Carro {
    // Atributos da classe Carro
    String marca;
    String cor;
    int ano;

    // Método da classe Carro
    void buzinar() {
        System.out.println("Beep! Beep!");
    }
}
```

```
// Criando a classe principal
public class Main {
    public static void main(String[] args) {
        // Criando um objeto meuCarro da classe Carro
        Carro meuCarro = new Carro();

        // Acessando atributos e métodos do objeto meuCarro
        meuCarro.marca = "Toyota";
        meuCarro.cor = "Vermelho";
        meuCarro.ano = 2020;

        System.out.println(meuCarro.marca); // Imprime: Toyota
        System.out.println(meuCarro.cor);   // Imprime: Vermelho
        System.out.println(meuCarro.ano);   // Imprime: 2020

        meuCarro.buzinar(); // Imprime: Beep! Beep!
    }
}
```

Criando um novo objeto carro na classe Carro:

```
Carro meuCarro = new Carro();
meuCarro.marca = "Toyota";
meuCarro.modelo = "Corolla";
meuCarro.ano = 2020;
```

Herança

A herança é um princípio fundamental da programação orientada a objetos que permite que as classes compartilhem atributos e métodos¹. Existem dois tipos principais de classes na herança:

- **Classe Base:** A classe que concede as características a uma outra classe.
- **Classe Derivada:** A classe que herda as características da classe base¹.

A herança é usada para reutilizar código ou comportamento generalizado e especializar operações ou atributos². Isso significa que uma classe derivada pode herdar atributos e métodos de uma classe base, além de adicionar novos atributos e métodos específicos da classe derivada³.

```

public class Pai {
    private String nome;

    public Pai(String nome) {
        this.nome = nome;
    }

    public String saudacao() {
        return "Olá, " + this.nome + "!";
    }
}

public class Filha extends Pai {
    private int idade;

    public Filha(String nome, int idade) {
        super(nome);
        this.idade = idade;
    }

    @Override
    public String saudacao() {
        return "Oi, " + super.getNome() + "! Você tem " + this.idade + " anos.";
    }
}

```

Criando uma Herança em nosso exemplo da classe carro:

```

public class CarroEsportivo extends Carro {
    int velocidadeMaxima;

    void turbo() {
        System.out.println("O turbo está ativado");
    }
}

```