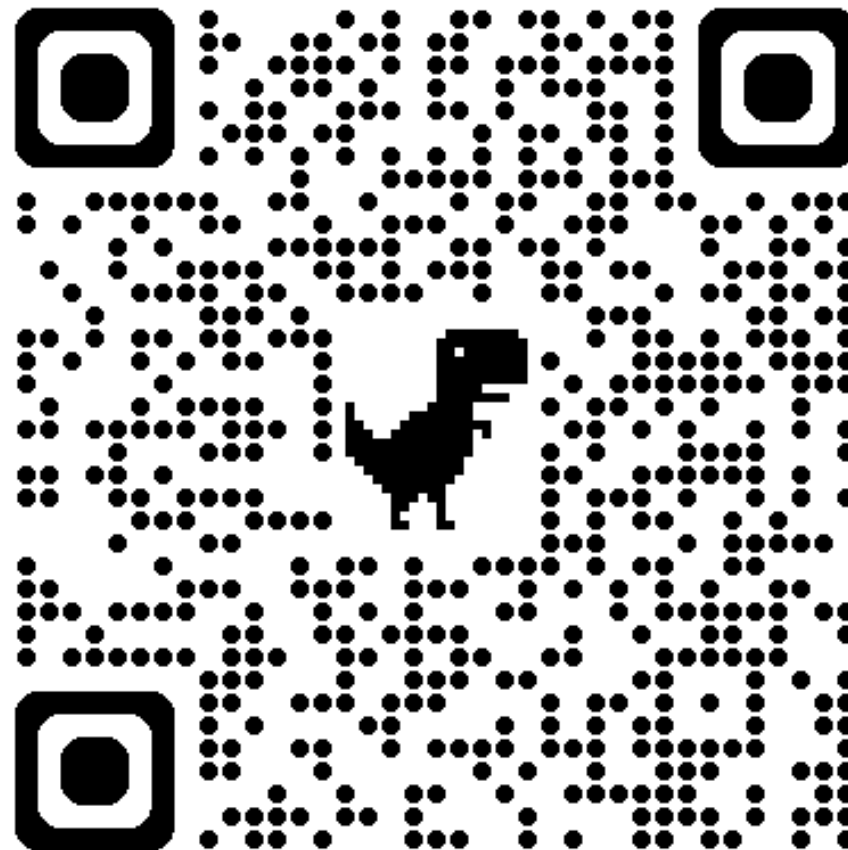


# PROGRAMAÇÃO ORIENTADA A OBJETOS

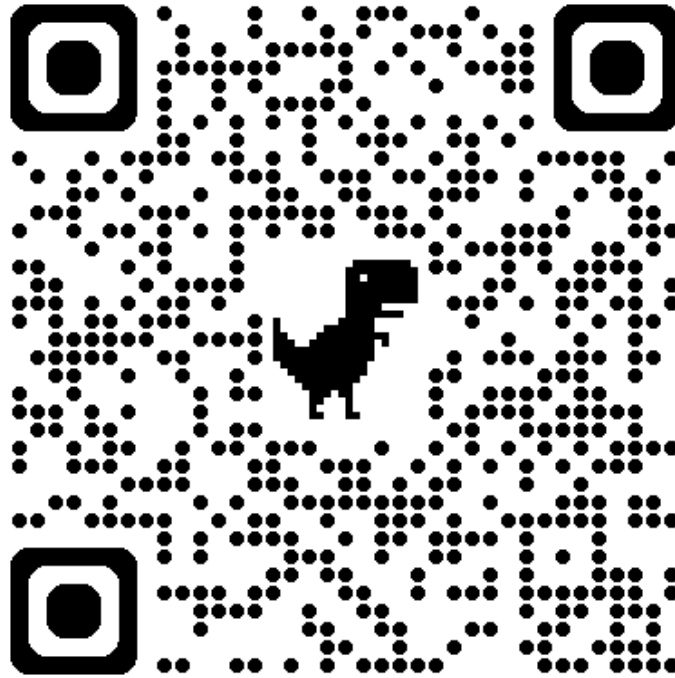
João Choma Neto

[joao.choma@unicesumar.edu.br](mailto:joao.choma@unicesumar.edu.br)

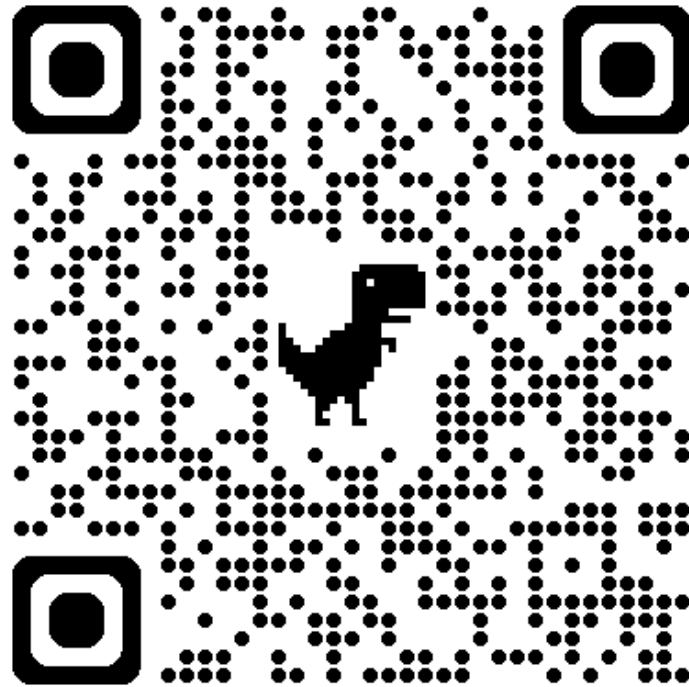
Unicesumar – Maringá



# IDE JAVA

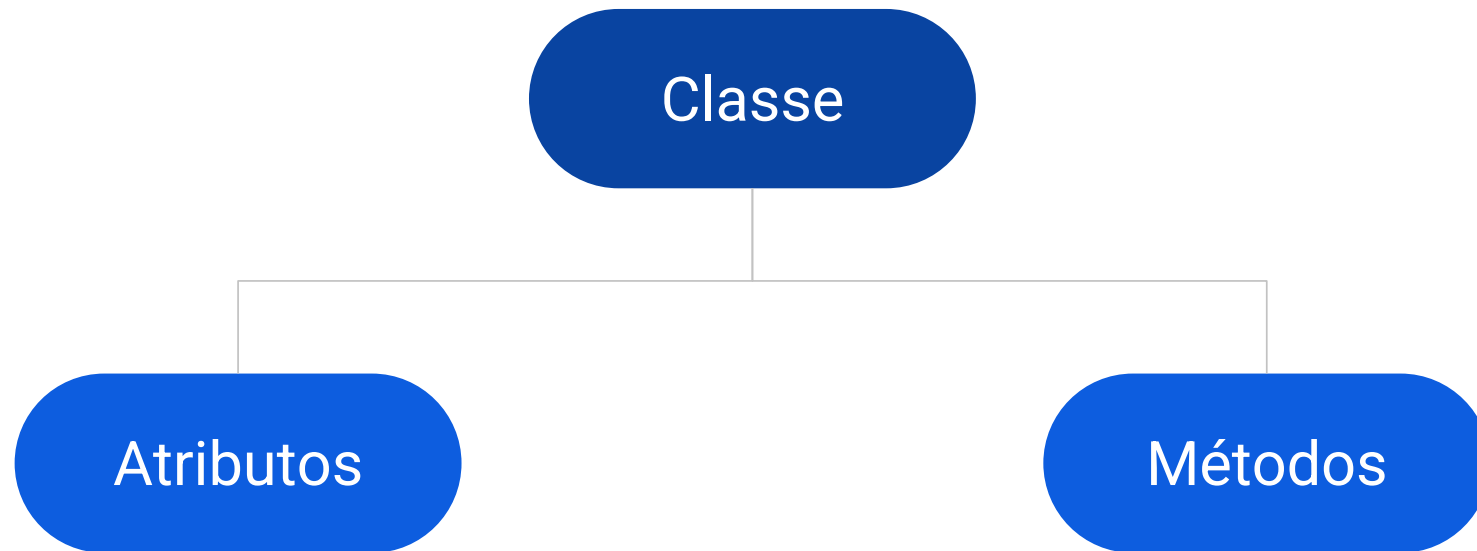


[https://www.jetbrains.com/idea/  
download/](https://www.jetbrains.com/idea/download/)



<https://www.oracle.com/br/java/technologies/downloads/>

# Classe



## Relacionamento entre classes

- **Associação:** é um relacionamento simples entre duas classes, em que uma classe utiliza a outra, mas não há uma dependência forte entre elas.

## Associação

- Uma classe Pessoa pode ter uma associação com uma classe Endereco, que representa o endereço dessa pessoa.

# Associação

```
public class Pessoa {  
    private String nome;  
    private int idade;  
    private Endereco endereco; // associação com a classe Endereco
```



# Associação

```
public class Endereco {  
    private String rua;  
    private int numero;  
    private String bairro;  
    private String cidade;  
    private String estado;
```

## Explicação do exemplo

- Nesse exemplo, a classe Pessoa tem uma associação com a classe Endereco, representada pelo atributo endereco.
- Essa associação permite que uma pessoa tenha um endereço associado a ela.
- A classe Endereco não depende da classe Pessoa e pode ser utilizada por outras classes também.

## Relacionamento entre classes

- **Herança:** é um relacionamento em que uma classe é derivada de outra classe, ou seja, uma classe mais específica é criada a partir de uma classe mais genérica.

## Relacionamento entre classes

- **Herança:** A classe mais específica herda os atributos e métodos da classe mais genérica e pode adicionar ou sobrescrever esses elementos.

## Herança

- . Por exemplo, uma classe Gato pode ser derivada de uma classe Animal, que representa qualquer tipo de animal.

## Herança

```
public class Animal {  
    private String nome;  
    private int idade;
```

Herança

```
public class Cachorro extends Animal {  
    private String raca;
```

# Herança

```
public class ExemploHeranca {  
    public static void main(String[] args) {  
        Cachorro cachorro = new Cachorro("Rex", 3, "Vira-lata");  
        System.out.println("Nome: " + cachorro.getNome());  
        System.out.println("Idade: " + cachorro.getIdade());  
        System.out.println("Raça: " + cachorro.getRaca());  
        cachorro.emitirSom(); // método da superclasse  
        cachorro.latir(); // método da subclasse  
    }  
}
```



## Explicação do exemplo

- Nesse exemplo, a classe Cachorro herda da classe Animal os atributos nome e idade e o método emitirSom.
- A classe Cachorro adiciona o atributo raca e o método latir.
- No exemplo de uso, um objeto da classe Cachorro é criado e são chamados seus métodos e os métodos herdados da classe Animal.

# PROGRAMAÇÃO ORIENTADA A OBJETOS

João Choma Neto

[joao.choma@unicesumar.edu.br](mailto:joao.choma@unicesumar.edu.br)

Unicesumar – Maringá

