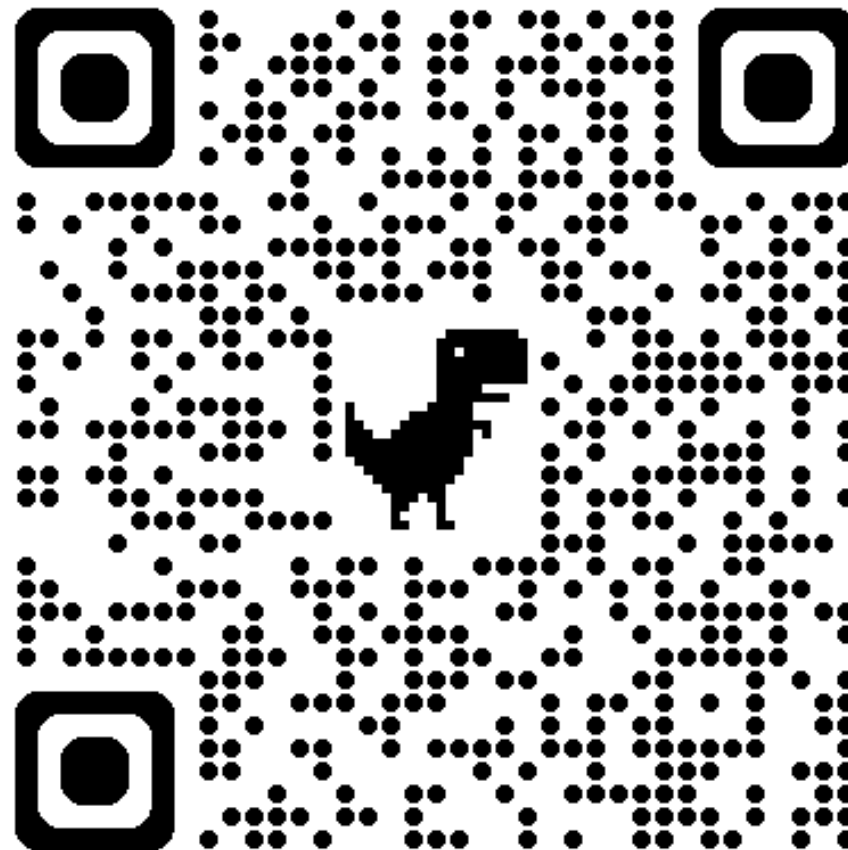


# PROGRAMAÇÃO ORIENTADA A OBJETOS

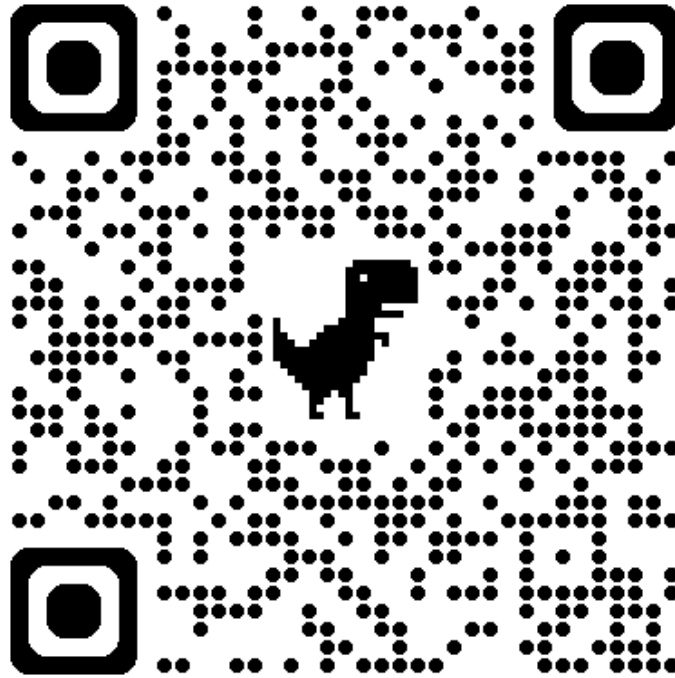
João Choma Neto

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

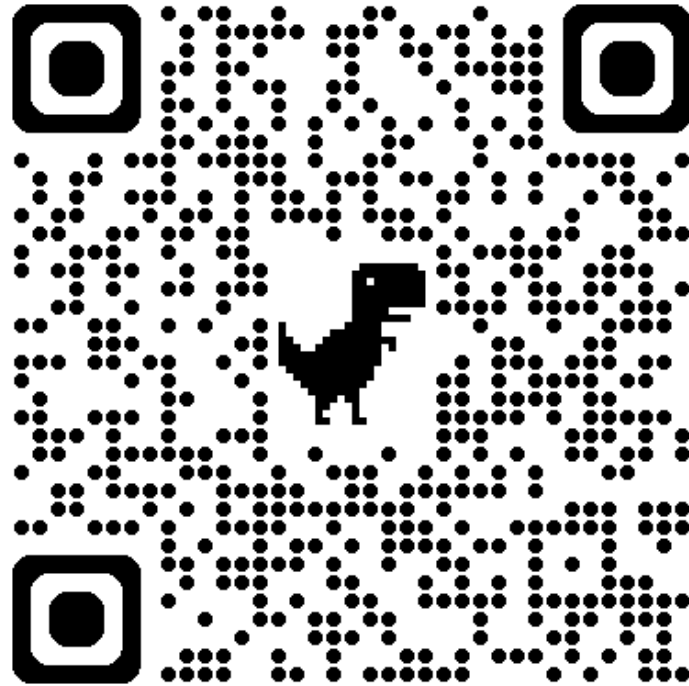
Unicesumar – Maringá



# IDE JAVA

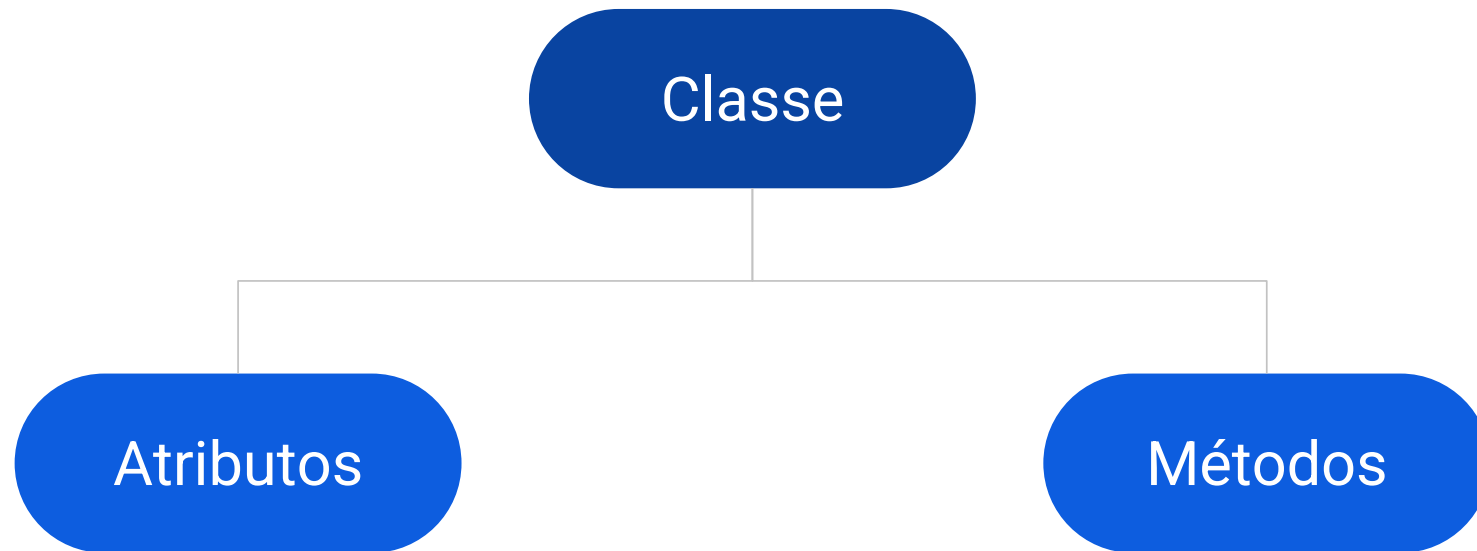


[https://www.jetbrains.com/idea/  
download/](https://www.jetbrains.com/idea/download/)



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

# Classe



# Atributos

Um **atributo** é uma variável que representa uma **característica** ou **propriedade** de um objeto

Cada objeto instanciado a partir de uma classe tem seus **próprios** valores para os atributos da classe

Eles **descrevem** as características dos objetos e podem ser usados para armazenar informações

# Classe Pessoa

- Atributos
- Métodos

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

# Classe Carro

## Modificadores de acesso

- private
- public

```
public class Carro {  
    private String modelo;  
    private int ano;  
    private double velocidade;  
  
    public Carro(String modelo, int ano) {  
        this.modelo = modelo;  
        this.ano = ano;  
        this.velocidade = 0.0;  
    }  
}
```



# Atributos

Um **atributo** é semelhante a uma **variável** em programação estruturada

Diferenças:

- Encapsulamento: Atributos em OO estão **encapsulados** dentro de objetos, enquanto as variáveis geralmente visíveis e acessíveis em todo o programa.
- Acesso: Atributos em OO podem ser acessados e modificados apenas por meio de **métodos** da classe, as variáveis em podem ser acessadas e modificadas em qualquer parte do programa.
- Tipos: Atributos em OO são **tipados**, o que significa que cada atributo tem um tipo de dados específico, enquanto as variáveis em programação estruturada podem ser tipadas ou não, dependendo da linguagem de programação.

# Classe Pessoa

## Atributos

- Nome (privado – acessado pelo método público)
- Idade (privado – acessado pelo método público)

## Métodos

- gets
- sets

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

# Métodos

Os **métodos** representam o **comportamento** do objeto

São modificadores de **estado** do objetos

São modificadores dos **atributos** definidos na classe

São responsáveis por **acessar** e modificar os atributos da classe

# Classe Carro

## Métodos

- acelerar (públicos)
- frear (públicos)

```
public void acelerar(double kmPorHora) {  
    this.velocidade += kmPorHora;  
}  
  
public void frear(double kmPorHora) {  
    this.velocidade -= kmPorHora;  
    if (this.velocidade < 0) {  
        this.velocidade = 0;  
    }  
}
```

O que temos em uma classe

- . Nome
- . Atributos
- . Método construtor – vazio ou não
- . Métodos get e set
- . Métodos de ação

Nome

```
public class Carro{  
  
}
```

# Atributos

```
private String modelo;  
private String marca;  
private int ano;  
private String cor;  
private int velocidadeAtual;  
private int velocidadeMaxima;
```

# Método construtor

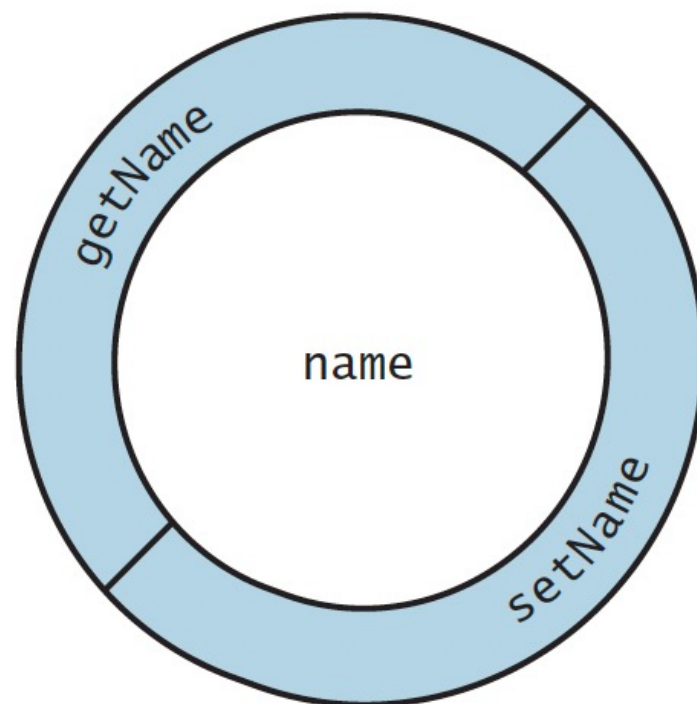
```
public Carro(String modelo, String marca, int ano, String cor, int
velocidadeMaxima) {
    this.modelo = modelo;
    this.marca = marca;
    this.ano = ano;
    this.cor = cor;
    this.velocidadeAtual = 0;
    this.velocidadeMaxima = velocidadeMaxima;
}
```



# Método Construtor

```
public Carro() {  
    this.modelo = modelo;  
    this.marca = marca;  
    this.ano = ano;  
    this.cor = cor;  
    this.velocidadeAtual = 0;  
    this.velocidadeMaxima = velocidadeMaxima;  
}
```

# Encapsulamento



# Gets e Sets

```
public String getModelo() {  
    return modelo;  
}
```

```
public String getMarca() {  
    return marca;  
}
```

```
public int getAno() {  
    return ano;  
}
```

# Gets e Sets

```
public String getCor() {  
    return cor;  
}
```

```
public int getVelocidadeAtual() {  
    return velocidadeAtual;  
}
```

```
public int getVelocidadeMaxima() {  
    return velocidadeMaxima;  
}
```

# Método de ação

```
public void acelerar(int incremento) {  
    if (this.velocidadeAtual + incremento <= this.velocidadeMaxima) {  
        this.velocidadeAtual += incremento;  
    } else {  
        this.velocidadeAtual = this.velocidadeMaxima;  
    }  
}
```

```
public void frear(int decremento) {  
    if (this.velocidadeAtual - decremento >= 0) {  
        this.velocidadeAtual -= decremento;  
    } else {  
        this.velocidadeAtual = 0;  
    }  
}
```

# Método de ação

```
public void acelerar(int incremento) {  
    if (this.velocidadeAtual + incremento <= this.velocidadeMaxima) {  
        this.velocidadeAtual += incremento;  
    } else {  
        this.velocidadeAtual = this.velocidadeMaxima;  
    }  
}
```

# Método de ação

```
public void frear(int decremento) {  
    if (this.velocidadeAtual - decremento >= 0) {  
        this.velocidadeAtual -= decremento;  
    } else {  
        this.velocidadeAtual = 0;  
    }  
}
```



# PROGRAMAÇÃO ORIENTADA A OBJETOS

João Choma Neto

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

Unicesumar – Maringá

