Análise e Desen. de Sistemas

POO

Programação Orientada a Objetos I

Prof. Letícia Pieper

60 h

UNIFASIPE
CENTRO UNIVERSITÁRIO

Fundamentos da POO

- Paradigma de Programação
- Baseado nos conceitos do mundo real
- **Mapear o mundo Real, no mundo computacional.**

o que é paradigma?

conjunto de formas vocabulares que servem de modelo para um sistema de flexão ou de derivação

#Fundamentos

Conceito de classe

Descrição de algo, projeto de algo.

Exemplo Carro:

*Propriedades
Fabricante
Modelo
Cor
Etc.

*Comportamentos
Ligar
Desligar
Frear
Etc.

#Fundamentos

Criando a Classe Carro

```
public class Carro {
    String fabricante;
    String modelo;
    String cor;
    int anoDeFabricacao;
    boolean ligado;
    int kmRodado = 0;
}
```

#Fundamentos

Criando objetos

```
public class ProgramaCarro {
    public static void main(String[] args){
        System.out.println("Olá");

        Carro uno = new Carro();
        uno.anoDeFabricacao = 2022;
        uno.cor = "Vermelho";
        uno.fabricante = "Fiat";
        uno.modelo = "Way";
    }
}
```


Criando o “Proprietário do Objeto carro.”

```
public class Entidade {
    String nome;
    String cpf;
    String idade;
    String logradouro;
    String bairro;
    String cidade;
}
```

#Fundamentos

Dentro da nossa class carro, vamos adicionar:

```
Entidade proprietario = new Entidade();
```


Vamos implementar nosso código:

```
public class IntroPOO {
    public static void main(String[] args){
        Carro meuCarro;
        meuCarro = new Carro();
}
```


```
public class IntroPOO {
    public static void main(String[] args){
        Carro meuCarro;
        meuCarro = new Carro();

        meuCarro.anoDeFabricacao = 1994;
        meuCarro.cor = "Vermelho";
        meuCarro.fabricante = "Volvo";
        meuCarro.modelo = "Caminhão";

       System.out.println(meuCarro.fabricante);
        System.out.println(meuCarro.modelo);

}
```

Adicionando dados do proprietário:

```
meuCarro.proprietario.nome = "Willian";
meuCarro.proprietario.cidade = "Sinop-MT";
meuCarro.proprietario.cpf = "039.933.561-79";
meuCarro.proprietario.logradouro = "Rua das Primaveras";
meuCarro.proprietario.bairro = "Jardim Jacarandas";

System.out.println(meuCarro.proprietario.nome);
System.out.println(meuCarro.proprietario.cpf);
```

Criando métodos: (Class Carro)

```
    void ligar() {
        if (ligado) {
            return;
        }
        System.out.println("Carro ligado");
        ligado = true;
    }
    void desligar() {
        if (!ligado) {
            return;
        }
        System.out.println("Carro desligado");
        ligado = false;
    }
```

Chamando métodos: (IntroPOO)

```
meuCarro.ligar();

meuCarro.desligar();
```

```
String resumoCarro(){
  return modelo + " - " + fabricante;
}

void rodar(int kmRodado){
  this.kmRodado = this.kmRodado + kmRodado;
}

  * Carro Class
```

Chamando métodos: (IntroPOO)

```
System.out.println(meuCarro.resumoCarro());
/* ------*/
System.out.println(meuCarro.kmRodado);
meuCarro.rodar(1000);
System.out.println(meuCarro.kmRodado);
```

Exercício:

Criando uma Classe

1. Levante os requisitos
2. Crie a Classe com os tipos levantados.
3. Implemente a classe, atribua dados, e plote os resultados.
4. Crie pelo menos 2 métodos (plote seus resultados)