



FLUTTER WITH VINI.

# AULA 9 -

# POLIMORFISMO.

# Roteiro.

- Sorteio.
- Revisão.
- Polimorfismo.
- Intervalo
- Quiz.
- Exercícios.
- Encapsulamento.
- Quiz.
- Exercícios.

# **POLIMORFISMO.**

*Conseguir alterar métodos  
para cada tipo de classe  
filha.*



**Imagine que  
a classe  
veiculo tenha  
o método  
mover.**

*Para cada tipo de veiculo se mover precisamos de diferentes ações.*

Agora solta a embreagem e acelera devagar



# Ações para mover um carro.

*Virar a chave.*

*Pisar no pedal da embreagem.*

*Mudar a marcha.*

*Soltar freio de mão.*

*Pisar no pedal do acelerador.*



# Ações para mover uma moto.

*Tirar o pezinho.*

*Virar a chave.*

*acionar a embreagem.*

*Trocar marcha pelo pedal.*

```
import 'dart:io';

class Veiculos {
  String nome;
  String marca;

  Veiculos({required this.nome, required this.marca});

  void mover(){
    print("$nome acelerando");
  }
}

class Carro extends Veiculos{

  Carro({required String nome, required String marca}):super(marca: marca,nome: nome);

  @override
  void mover() {
    print(" $nome Vire a chave, pise na embreagem , mude a marcha, desative o freio de mao e pise no pedal de acelerador");
  }
}

class Moto extends Veiculos{
  Moto({required String nome, required String marca}):super(marca: marca,nome: nome);

  @override
  void mover(){
    print("$nome Vire a chave,  acione a embreagem , pise no pedal da marcha, acelere  ");
  }
}
```

```
Run | Debug  
38 void main() {  
39     Carro uno = Carro(marca: "Fiat ", nome: "Unin");  
40     uno.mover();  
41  
42     Moto Biz = Moto(marca: "Honda ", nome: "Biz");  
43     Biz.mover();  
44 }
```

PROBLEMS 19    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

- PS C:\Users\viinha\OneDrive\Desktop\excel> dart .\polimorfismo.dart  
Unin Vire a chave, pise na embreagem , mude a marcha, desative o freio de mao e pise no pedal de acelerador  
Biz Vire a chave, acione a embreagem , pise no pedal da marcha, acelere

# **Exercício 1.**

**CRIE UMA CLASSE ANIMAL COM O MÉTODO FALAR().  
IMPLEMENTE SUBCLASSES CACHORRO, GATO E PAPAGAIO, CADA UMA REDEFININDO FALAR() DE FORMA DIFERENTE.  
NO MAIN, FAÇA UMA LISTA DE ANIMAL E PERCORRA COM FOREACH, CHAMANDO FALAR() PARA CADA UM.**

# **Exercício 2.**

**Implemente a classe Pagamento com o método processar().  
Crie as subclasses Pix, CartaoCredito e Boleto, sobrescrevendo processar()  
de forma distinta.  
No main, simule a escolha do usuário para selecionar o tipo de pagamento e  
chame o método correto.**

# **Exercício 3.**

**Crie a classe Forma com o método calcularArea().**

**Implemente Quadrado, Retangulo e Circulo, cada um com sua fórmula.**

**No main, faça uma lista de Forma e calcule a área de cada objeto usando o polimorfismo.**



FLUTTER WITH VINI.

# AULA 10 - ENCAPSULAMENTO.

# **ENCAPSULAMENTO.**

*Conseguir consultar e  
alterar valores dos atributos.*

*Estando em arquivos  
diferentes.*

# Quando usar GET?

**Imagine que você tem uma classe Carro, e você precisa pegar o valor da velocidade.**

**Após isso precisa colocar em uma outra variável e multiplicar por 2.**

**Desta forma usamos GET para colocar os valores dos atributos em uma variável.**

# get.dart

```
class Carro {  
    int _velocidade = 50; // atributo privado  
  
    String nome = "";  
  
    // Getter  
    int get receber_velocidade {  
        return _velocidade;  
    }  
  
    Carro({required nome});  
}
```

# main.dart

```
import 'dart:io';  
import 'get.dart';  
  
Run | Debug  
void main() {  
    Carro uno = Carro(nome: "Uno");  
    // Usando GET para pegar o valor  
    int resultado = uno.receber_velocidade * 2;  
    print("Velocidade dobrada: $resultado km/h");  
}
```

# Quando usar SET?

**Imagine que você tem uma classe  
Carro, e você precisa alterar a  
velocidade.**

**Mas a velocidade não pode ser  
menor que 0.**

**Desta forma usamos para validar  
modificações nos atributos.**

# set.dart

```
import 'dart:io';

class Carro{
    String modelo;
    int _velocidade = 0; //Atributo privado precisa ter um valor inicial.
    //pois não consigo alterar diretamente.
    Carro({required this.modelo});

    //Criando função para ver o valor privado.
    int get get_velocidade {
        return _velocidade;
    }

    //Criando função que possibilita alterar o valor.
    set alterar_velocidade(int v){
        if(v < 0){
            print("Velocidade não pode ser menor que 0");
        }else{
            _velocidade = v;
        }
    }
}
```

# main.dart

```
import 'dart:io';
import 'encapsulamento.dart';

Run | Debug
void main() {
    Carro Carro1 = Carro(modelo: "Ferrari");
    Carro1.alterar_velocidade = -20; //Método para eu alterar.
    print(Carro1.get_velocidade); //Método para eu ver.

}
```

Velocidade não pode ser menor que 0



**GET** → quando você consulta valores de atributos privados.

**SET** → quando você altera valores de atributos privados com validação.

### **Exercício 1 – Controle de temperatura**

**Crie uma classe Termometro com:**

**Um atributo privado \_temperatura.**

**Um getter que devolve a temperatura em Celsius.**

**Um setter que:**

**Só aceite valores entre -50 e 100.**

**Caso o valor seja inválido, mostre uma mensagem de erro.**

**No main, crie um termômetro, defina uma temperatura válida e uma inválida, e exiba o valor lido.**



## **Exercício 2.**

**Crie uma classe Produto com:  
Atributo privado \_quantidade.**

**Getter estoque que retorna a quantidade disponível.**

**Setter adicionar para aumentar o estoque (não pode receber valores negativos).**

**Setter remover para retirar itens (não pode permitir que o estoque fique negativo).**

**No main, crie um produto com nome e estoque inicial.**

**Faça operações de entrada e saída de estoque e mostre o resultado no console.**





FLUTTER  
COM  
VINICIUS.

VLW!