

06 - Trabalhando com POO - Classes

Classes em Dart

Em Dart, as classes são fundamentais para implementar POO, permitindo a criação de estruturas de código mais organizadas, reutilizáveis e fáceis de manter. Com classes, você pode representar entidades do mundo real ou conceitos abstratos, definindo propriedades (atributos) e comportamentos (métodos).

Para criar uma classe em Dart, utiliza-se a palavra-chave `class` seguida pelo nome da classe:

```
class Pessoa {  
    // Propriedades e métodos da classe  
}
```

As propriedades são `variáveis` que armazenam o estado de um objeto. Elas podem ser públicas ou privadas.

```
class Pessoa {  
    String nome;  
    int idade;  
  
    Pessoa(this.nome, this.idade);  
}  
  
// Criando um objeto  
void main() {  
    var pessoa = Pessoa('Bruno', 31);  
    print('Nome: ${pessoa.nome}, Idade: ${pessoa.idade} // Saída: Nome:  
Bruno, Idade: 31');  
}
```

Construtores

Construtores são métodos especiais usados para criar e inicializar objetos.

Se não especificarmos um construtor no Dart, ele criará um construtor padrão.

Isso não significa que veremos um construtor padrão gerado em sua classe. Em vez disso, ao criar uma nova instância de sua classe, esse construtor será chamado. Ele não terá argumentos e chamará o construtor da superclasse, sem argumentos também.

Construtores padrão

```
class Carro {  
    String fabricante;  
    String modelo;
```

```
String anoFabricacao;
bool comABS;
}

// declarando um construtor em uma classe
class Carro {
    String fabricante;
    String modelo;
    String anoFabricacao;
    bool comABS;

    Carro(String fabricante, String modelo, int ano, bool comABS) {
        this.fabricante = fabricante;
        this.modelo = modelo;
        this.anoFabricacao = ano;
        this.comABS = comABS;
    }
}

//ou podemos simplificar a sintaxe do Dart para realizar a tarefa.
class Carro {
    String fabricante;
    String modelo;
    String anoFabricacao;
    bool comABS;

    Carro(this.fabricante, this.modelo, this.anoFabricacao, this.comABS);
}
```

Construtores nomeados em Dart

Em dart é possível dar nomes diferentes aos seus construtores permitindo que sua classe tenha muitos construtores e represente melhor seus casos de uso fora da classe.

```
// O construtor semABS inicializa a variável de instância comABS como
false,
// antes que o corpo do construtor seja executado. Isso é conhecido como
// uma lista de inicializadores e você pode inicializar diversas
variáveis,
// separadas por uma vírgula.
class Carro {
    String fabricante;
    String modelo;
    String anoFabricacao;
    bool comABS;

    Carro(this.fabricante, this.modelo, this.anoFabricacao, this.comABS);

    Carro.semABS(this.fabricante, this.modelo, this.anoFabricacao): comABS
```

```
= false;  
}
```

Construtores "de fábrica" Um construtor de fábrica é um construtor que pode ser usado quando desejamos necessariamente que um construtor crie uma instância de sua classe.

Isso pode ser útil se queremos manter instâncias da nossa classe na memória e não querer criar uma outra a cada vez (ou se a operação de criação de uma instância for cara).

Outro caso de uso é se tivermos determinada lógica em nosso construtor para inicializar um campo final que não pode ser feito na lista inicializadora.

```
class Carro {  
    String fabricante;  
    String modelo;  
    String anoFabricacao;  
    bool comABS;  
  
    factory Carro.ford(String modelo, String anoFabricacao, bool comABS) {  
        return CarroFord(modelo, anoFabricacao, comABS);  
    }  
}  
  
class CarroFord extends Carro {  
    CarroFord(String modelo, String anoFabricacao, bool comABS):  
    super("Ford", modelo, anoFabricacao, comABS);  
}
```