

Modificadores

Modificador static

Os modificadores Static servem para que atributos e/ou métodos de uma classe não precisarem ser instanciadas para que sejam usadas no programa.

Assim deixando já na memória do computador.

Possibilitando também que os métodos sejam influenciados por atributos por atributos static na sua estrutura.

Sendo muito utilizado para ter um padrão a um objeto.

Exemplo utilizado para atributo static para deixar padrão um valor de uma classe.

```
class Pessoa{  
  
    static double alturaPadrao;  
  
    String nome = '';  
  
    double altura = alturaPadrao;  
  
}  
  
main{  
  
    Pessoa.alturaPadrao = 1.80;  
  
    Pessoa eu = Pessoa();  
  
    print('Minha altura padrão é ${eu.altura}')
```

Assim depois que foi definido apartir disso todos os objetos que serao instanciados iram utilizar esse valor padrão.

Modificador late

Esse tipo de modificador serve para que um atributo seja obrigatório para a sua instancia mas que não precise ser criado com um valor inicial e nem que seja

atribuido valor atraves de cosntrutor.

Assim deixando para depois esse atributo seja preenchido em tempo de execução e não em tempo de compilação, por exemplo de uma variável pré definida no escopo da classe.

O unico problema que pode ocorrer quando usamos esse tipo de operador é a exceção de ("LateInitializationError")

-Erro de não inicialização de variável.

Esse tipo de modificador é para evitar a utilização de uma variável do tipo nullable que pode acontecer problemas muito piores e deixando o escopo mais feio quando precisa ser utilizado essa variável

Exemplo de como é chamado Late Inicializate (execução tardia)

Usado em um atributo

```
class Pessoa{
    Pessoa({required this.nome})

    String nome;
    Late String cpf;
}
```

Usado em um método

```
class Pessoa{
    Pessoa({required this.nome})

    String nome;

    Late double temperaturo = medir();

    double medir(){
        return 37.00;
    }
}
```

Operador ? (Ponto de interrogação)

Esse tipo de modificador é utilizado denominar o tipo que pode ser nullable.

Assim sendo também um modificador para se verificar se a variável é nulla.

```
if(nome??){  
    return 'Essa variável é nulla'  
}  
else{  
    return 'Essa variável não é nulla'  
}
```