

Kotlin – Construtores

O que é um construtor?

É uma função que tem o mesmo nome da classe e a sua única atribuição é criar uma nova instância desta classe.

Como fazemos para criar uma instância de uma classe?

Na linguagem *Java*, usamos a palavra reservada ***new*** antecedendo a chamada do método construtor:

```
Carro carro = new Carro();
```

Já em *Kotlin*, a chamada é muito mais simples. Pois, usamos o construtor diretamente sem a necessidade do ***new***:

```
val carro = Carro()
```

Quantos tipos de construtores existem e quais são?

Em *Kotlin*, existem dois tipos:

Primário

Ele segue o nome da classe e faz parte do cabeçalho da mesma. Só existe apenas um construtor primário por classe, independentemente que seja declarado com acesso público ou privado.

Se todos os parâmetros possuírem um valor padrão, o compilador irá gerar novo um construtor sem parâmetros adicionais que utilizará os valores padrões.

Secundário

É declarado com a palavra reservada ***constructor***, dentro do corpo da classe, sendo possível declarar um número ilimitado destes construtores.

Porém, em *Kotlin*, não é indicado realizar sobrecarga desses construtores. Já que é interessante usar o recurso da linguagem para atribuir os valores padrões que irão cobrir diversas possibilidades para instanciar um novo objeto.

Exemplos:

O construtor sem parâmetros pode ser declarado de várias maneiras, mas a palavra reservada ***constructor*** poderá ser omitida caso não tenha nenhum modificador de visibilidade ou uma anotação associada ao construtor primário:

```
class Carro
    class Carro()
    class Carro constructor()
    class Carro public @Inject constructor()
```

Se não existir nenhum construtor declarado explicitamente em uma classe, o construtor ***default***, sem argumentos, será gerado automaticamente:

```
class Carro
```

Em seguida, vamos visualizar um construtor primário com parâmetros:

```
class Carro(marca: String, modelo: String, ano: Int)
```

Também é possível criar uma sobrecarga do construtor supracitado, simplesmente atribuindo um valor padrão para uma ou mais variáveis:

```
class Carro(marca: String, modelo: String = "Ford Mustang", ano: Int = 1967)
```

E por fim, para declarar um construtor secundário, faz-se necessário uma chamada direta a ***this*** ou precisaremos delegar essa responsabilidade para outro construtor que faça essa referência:

```
class Carro(nome: String) {
    private var ano: Int = 1967

    constructor(modelo: String, ano: Int): this(modelo) {
        this.ano = ano
    }
}
```

Por hoje é só pessoal, sucesso nos estudos!