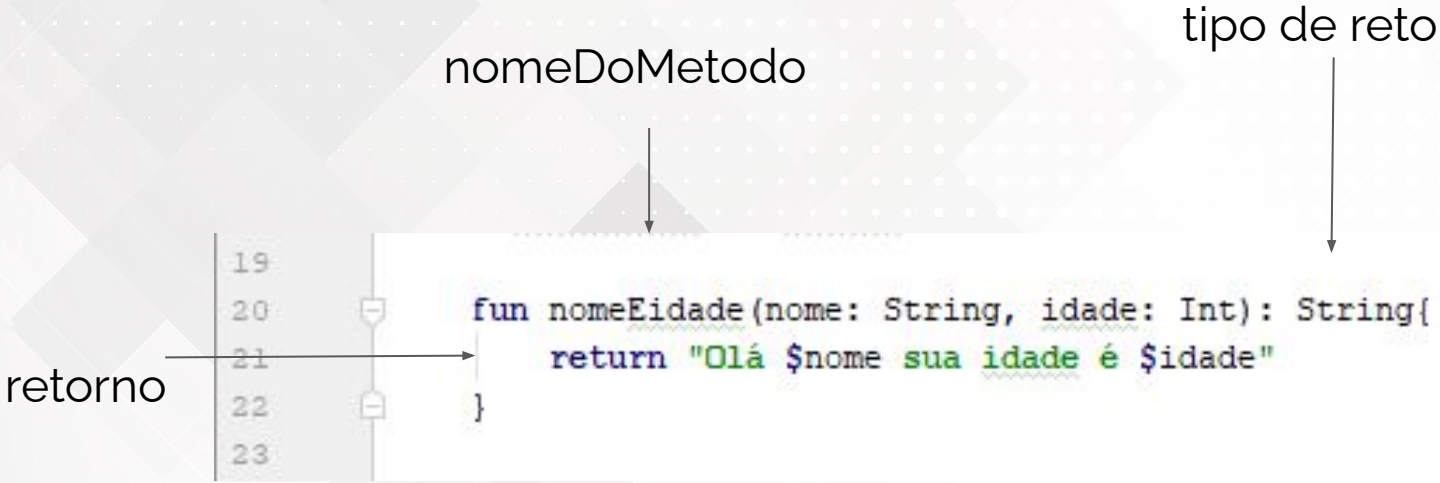
Aula 14

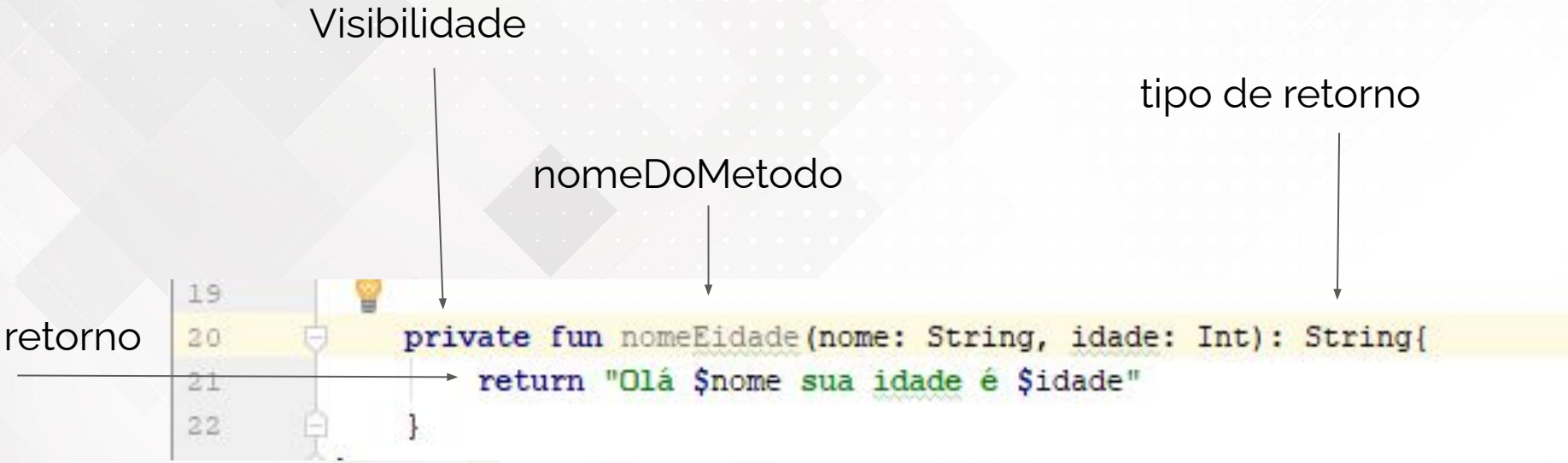
Functions parte I

DigitalHouse >
Coding School

Assinatura de funções - Exemplo 1



Assinatura de funções - Exemplo 2



Funções- Default arguments

Os parâmetros de função podem ter valores padrão além disso os valores padrão são definidos usando o tipo = juntamente com o valor.

```
20
21      fun nome(nome: String = ""){
22
23      }
24  }
25
```

Funções- Named arguments

Os parâmetros de função podem ser nomeados. Isso é muito conveniente quando uma função possui um grande número de parâmetros.

```
24
25      fun dadosPessoais(nome: String = "",
26                        sobrenome: String = "",
27                        idade: Int,
28                        anoNascimento: Int,
29                        altura: Double,
30                        peso: Double,
31                        rg: String,
32                        cpf: String) {
33      }
34  }
35
```

Funções- Named arguments (uso)

```
1  fun main() {
2
3      val pessoa = Pessoa()
4
5      pessoa.dadosPessoais(nome = "Jess",
6                           sobrenome = "Corrêa",
7                           idade = 18,
8                           anoNascimento = 2001,
9                           altura = 1.70,
10                          peso = 50.0,
11                          rg = "000000",
12                          cpf = "1111111")
13
14 }
15
16
```

Lambdas

Funções Lambda nada mais são do que funções anônimas, que não estão presas a um identificador, podendo serem passadas para ou retornadas em uma função.

```
val sum: (Int, Int) -> Int = { x: Int, y: Int -> x + y }
```

Lambdas

Uma expressão lambda é sempre cercada por chaves, as declarações de parâmetro na forma sintática completa ficam dentro de chaves e possuem anotações de tipo opcionais, o corpo segue um `->` sinal. Se o tipo de retorno inferido do lambda não for Unit, a última expressão (ou possivelmente única) dentro do corpo do lambda será tratada como o valor de retorno.

```
val sum = { x: Int, y: Int -> x + y }
```

Lambdas - it

É muito comum que uma expressão lambda tenha apenas um parâmetro.

Se o compilador puder descobrir a própria assinatura, é permitido não declarar o único parâmetro e omitir o `->`

O parâmetro será declarado implicitamente sob o nome it:

```
ints.filter { it > 0 } // this literal is of type '(it: Int) -> Boolean'
```

Exercícios