**Desenvolvendo aplicativos Android com a linguagem Kotlin**

//função principal teste

fun main() {

var nome = "Escudero"

var idade = 37

println("Meu nome é $nome e tenho $idade anos!")

}

--

//verificar o tamanho mínimo e máximo dos tipos de dados das variáveis

fun main() {

println("Double Max: " + Double.MAX\_VALUE + " - Double Min: " + Double.MIN\_VALUE)

println("Float Max: " + Float.MAX\_VALUE + " - Float Min: " + Float.MIN\_VALUE)

println("Long Max: " + Long.MAX\_VALUE + " - Long Min: " + Long.MIN\_VALUE)

println("Int Max: " + Int.MAX\_VALUE + " - Int Min: " + Int.MIN\_VALUE)

println("Short Max: " + Short.MAX\_VALUE + " - Short Min: " + Short.MIN\_VALUE)

println("Byte Max: " + Byte.MAX\_VALUE + " - Byte Min: " + Byte.MIN\_VALUE)

}

**Criando funções e seus benefícios**

Uma função permite economia de código e facilidade na hora de pequenos ajustes.

**Ex1:**

//criando função calculaBonus

//função sem retorno, apenas execução

fun calculaBonus(a: Int, b: Int, c: Int){

println("O bônus é: ${a + b + c}")

}

//uso da função durante o código

fun main() {

val a = 10

val b = 20

val c = 30

calculaBonus(a,b,c)

//

//

//

calculaBonus(a,b,c)

//

//

calculaBonus(a,b,c)

}

//obs.: se depois de um tempo for necessário mudar algo na regra de bônus, basta alterar a função calculaBonus

**Ex2:**

//criando função hello

//função com retorno

fun hello(nome:String):String{

return "Olá, $nome"

}

//utilizando a função hello dentro de outra função

fun main() {

println(hello(nome = "Escudero"))

println("Seja bem vindo!")

}

**Simplificando funções em apenas uma linha**

//somente funções com apenas uma instrução

**Ex1:**

//função original

fun hello(nome:String):String{

return "Olá, $nome"

}

//função resumida com apenas uma linha

fun hello(nome:String) = "Olá, $nome"

**Ex2:**

fun soma (a:Int,b:Int) = a + b

fun main() {

println("O resultado da soma é ${soma(2,2)}")

}

**Incremento:**

numero + 1 = numero++

numero++ utiliza o valor e depois soma +1

++numero soma +1 antes de utilizar o valor

**Ex:**

//incremento

fun main() {

var numero = 10

println(numero)

numero++

println("numero++ = ${numero++}")

numero--

println("numero-- = ${numero--}")

++numero

println("++numero = ${++numero}")

--numero

println("--numero = ${--numero}")

numero += 2

println("numero + 2 = $numero")

numero -= 2

println("numero - 2 = $numero")

numero \*= 2

println("numero \* 2 = $numero")

numero /= 2

println("numero / 2 = $numero")

numero %= 3 //sobra/resto da divisão

println("numero%= 3 = $numero")

}

**Controle de Fluxo**

**Ex1:**

//Controle de Fluxo

fun calculaMedia(media: Float) {

if (media >= 7) {

println("Parabéns, sua média é $media e você foi aprovado!")

} else {

println("Que pena, sua média é $media e você foi reprovado!")

}

}

fun main() {

val a: Float = 7f

val b: Float = 5f

val c: Float = 6f

calculaMedia(media=(a+b+c)/3)

}

**Ex2:**

//Controle de Fluxo

fun calculaBonus(cargo:String, salario:Float): Float{

//Gerente, Coordenador e Líder

val bonus:Float

if (cargo == "Gerente"){

bonus = salario \* 0.5f

} else if (cargo == "Coordenador"){

bonus = salario \* 0.3f

}else{

bonus = salario \* 0.1f

}

//tem que ter retorno senão dá erro

return bonus

}

fun main() {

println("Gerente, seu bônus será de: " + calculaBonus(cargo = "Gerente", salario = 2000f))

println("Coordenador, seu bônus será de: " + calculaBonus(cargo = "Coordenador", salario = 1500f))

println("Líder, seu bônus será de: " + calculaBonus(cargo = "Líder", salario = 1000f))

}

**Operador Ternário:**

//Operador Ternário

fun main() {

val valor = 10

//Java

//String str = valor == 10 ? "Sim" : "Não";

//Kotlin - utiliza-se if e else

val str = if (valor == 10) "Sim" else "Não"

println(str)

}

**Conjunção && (e)**

//validação de 2 condições

fun main() {

if (exp1 && exp2) {

//Código

}

}

| **exp1** | **exp2** | **resultado** |
| --- | --- | --- |
| true | false | false |
| false | true | false |
| false | false | false |
| true | true | true |

**Disjunção|| (ou)**

//validação de 2 condições

fun main() {

if (exp1 || exp2) {

//Código

}

}

| **exp1** | **exp2** | **resultado** |
| --- | --- | --- |
| true | false | true |
| false | true | true |
| false | false | false |
| true | true | true |

**Negação !**

//negação !

fun main() {

if (!exp1) {

//Código

}

}

**Validação Múltiplos Tipos**

//multiplos tipos

// **e** tem prioridade em relação ao **ou**

fun main() {

if ((exp1 && exp2) || exp3 || exp4) {

//Código

}

}

| **exp1 && exp2** | **exp3** | **exp4** | **resultado** |
| --- | --- | --- | --- |
| true | false | false | true |
| false | true | false | true |
| false | false | true | true |

**Elvis Operator ?:**

//verifica se a variável tem valor null e em caso positivo atribui o valor informado após ?:

fun main() {

val test: Int? = null

val op: Int = test ?: 100 // op recebe teste se teste for null, recebe 100

println(op)

}

**Controle de Fluxo - When**

**Ex1:**

//uso do when

fun operacao (a: Int, b: Int, c: String){

val res: Int

when(c){

"som" -> {

res = a + b

println("O resultado da soma de $a + $b é $res")

}

"sub" -> {

res = a - b

println("O resultado da subtração de $a - $b é $res")

}

"mult" -> {

res = a \* b

println("O resultado da multiplicação de $a x $b é $res")

}

"div" -> {

res = a / b

println("O resultado da divisão de $a / $b é $res")

}else ->{

println("Você digitou uma operação invalida")

}

}

}

fun main() {

operacao(10,10,"som")

operacao(10,10,"sub")

operacao(10,10,"mult")

operacao(10,10,"div")

operacao(10,10,"teste")

}