

Faça como eu fiz

Nessa aula aprendemos a criar variáveis do tipo inteiro e como compará-las utilizando uma estrutura de `if` s aninhados:

```
val idade1 = 25
val idade2 = 19
val idade3 = 33
val idade4 = 20

val maiorIdade = if (idade1 > idade2 && idade1 > idade3 && idade1 > idade4) {
    idade1
} else if (idade2 > idade3 && idade2 > idade4) {
    idade2
} else if (idade3 > idade4) {
    idade3
} else {
    idade4
}
```

Em seguida vimos que utilizar variáveis comuns não é o ideal para fazer esse tipo de comparação. Um estrutura mais própria para essa tarefa seria a utilização de arrays. Então modificamos o código anterior para a nova versão:

```
val idades = IntArray(4)
idades[0] = 25
idades[1] = 19
idades[2] = 33
idades[3] = 20

val maiorIdade = if (idades[0] > idades[1] && idades[0] > idades[2] && idades[0] > idades[3]) {
    idades[0]
} else if (idades[1] > idades[2] && idades[1] > idades[3]) {
    idades[1]
} else if (idades[2] > idades[3]) {
    idades[2]
} else {
    idades[3]
}
```

Essa é uma forma funcional de declarar um array de idades, mas existe uma forma mais simples e mais concisa de fazer a mesma coisa:

```
val idades: IntArray = intArrayOf(25, 19, 33, 20)
```

Apesar de deixarmos o código um pouco mais dinâmico dessa forma, ainda não resolvemos o problema dos `if` s aninhados. Vimos então que, para resolver o problema de `if` s aninhados, precisamos utilizar uma estrutura de repetição para iterar por todos os elementos do array sem precisarmos descrever um por um, como exemplo de estrutura de repetição, utilizamos o laço `for`:

```
var maiorIdade = Int.MIN_VALUE
for (idade in idades) {
    if(maiorIdade < idade){
        maiorIdade = idade
    }
}
```

Dessa forma a gente conseguiu reduzir muito a complexidade do nosso sistema, de forma que, caso a gente precise adicionar mais uma idade no nosso array, não seria necessário alterar o algoritmo que identifica qual é a maior idade:

```
val idades: IntArray = intArrayOf(25, 19, 33, 20, 55, 40)
```

Usando um algoritmo bem parecido, a gente conseguiu identificar também qual é a menor idade do array:

```
var menorIdade = Int.MAX_VALUE
idades.forEach { idade ->
    if(idade < menorIdade) {
        menorIdade = idade
    }
}
```

Depois disso, a gente viu como alterar arrays para adicionar 10% de aumento no salário dos funcionários de uma empresa:

```
val salarios: DoubleArray = doubleArrayOf(1500.50, 2300.0, 5000.0, 8000.0, 10000.0)
val aumento = 1.1
for(indice in salarios.indices) {
    salarios[indice] = salarios[indice] * aumento
}
```

Vimos também que podemos utilizar o `forEachIndexed` para para iterar pelo array e alterar os valores de seus elementos:

```
salarios.forEachIndexed { i, salario ->
    salarios[i] = salario * aumento
}
```