

Mobile Application Development

Prof. Fernando Pinéo

Introdução a linguagem de programação Kotlin Parte III

Interfaces

O que é uma interface..?

```
/*
```

```
Podemos dizer que a interface é um contrato que quando  
assumido por uma classe,deverá ser implementado.
```

```
Interface é utilizado pois podemos ter muitos objetos(classes  
que podem possuir a mesma ação(métodos), porém, podem  
executá-las de maneira de diferente..|
```

```
*/
```

Vamos criar algumas classes...

Criando classe Jose e Fernando, e em seguida estendendo da classe Cidadao..

FAP

```
open class Cidadao{  
    fun direitosDeveres() {  
        println("Todo cidadao tem direitos e deveres..")  
    }  
}  
  
class Jose:Cidadao(){  
  
}  
  
class Fernando:Cidadao(){  
  
}
```

Criando nossa interface...

```
interface Professor{  
    fun passarConcurso()  
}
```

Utilizando interface na classe Jose..

```
class Jose:Cidadao(),Professor{  
    override fun passarConcurso() {  
        println("Passou em concurso em RN")  
    }  
}
```


Criando objeto jose..



```
fun main() {
```

```
    val jose = Jose()
```

```
    jose.passarConcurso()
```

```
}
```

Collections - ArrayList

Definicação de Collection..

```
/*  
    Collections ou coleções são implementações de  
    estruturas de dados  
    -List -> Imutável, tamanho fixo, apenas para leitura  
    💡 -ArrayList -> Mutável, sem tamanho fixo, pode adicionar  
    ou remover  
*/
```

listOf – Collection – Imutável, por este motivo é apresentado o erro na fig abaixo.., estamos tentando add BA na lista..

FLAP

```
fun main() {  
    ! var listItens = listOf("SP", "RJ", "RN")  
    listItens.add("BA")  
}
```

Build Output

Build untitled2: failed At 21/08/2024 08:58 with 2 errors 468 ms

! :compileKotlin 1 error 210 ms

▼ Main.kt src\main\kotlin 1 error

! Unresolved reference 'add'. :12

! Compilation error

e: [file:///C:/Us](#)

arrayListOf, já consigo realizar alterações em nossa collection..

FIAP

```
fun main() {  
    var listItens = arrayListOf("SP", "RJ", "RN")  
    listItens.add("BA")  
    listItens.remove(element: "SP")  
    listItens.removeAt(index: 1)  
  
    println(listItens.size)  
    println(listItens.isEmpty())  
}
```

Collections – Set e Map

Definição de SET e MAP

```
/*  
    SET -> Não permite elementos duplicados, é mais rápido  
    💡 que o ArrayList, não mantém a ordem dos elementos  
    MAP -> é uma implementação chave/valor  
*/
```

Collections com Set..

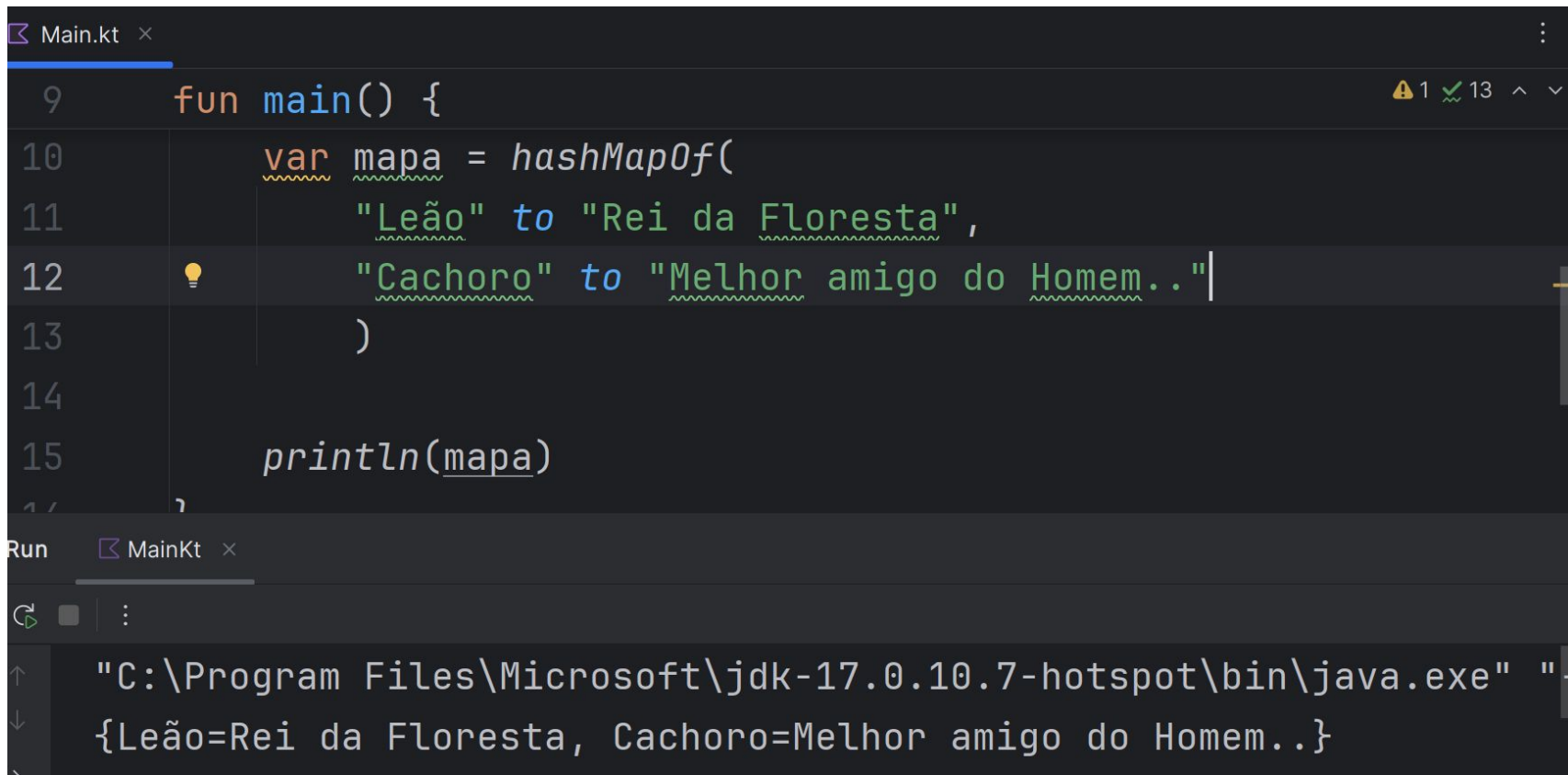
```
fun main() {  
    var itens = HashSet("Caio", "Gabriel", "Elen", "Caio")  
    println(itens)  
}
```

MainKt x

"C:\Program Files\Microsoft\jdk-17.0.10.7-hotspot\bin\java.exe"
[Elen, Caio, Gabriel]

No exemplo acima, foi criado uma collection com 04 nomes, porém ao printar a variável **itens**, é possível observar que o nome duplicado foi removido, permanecendo somente um.

Collection com Map..

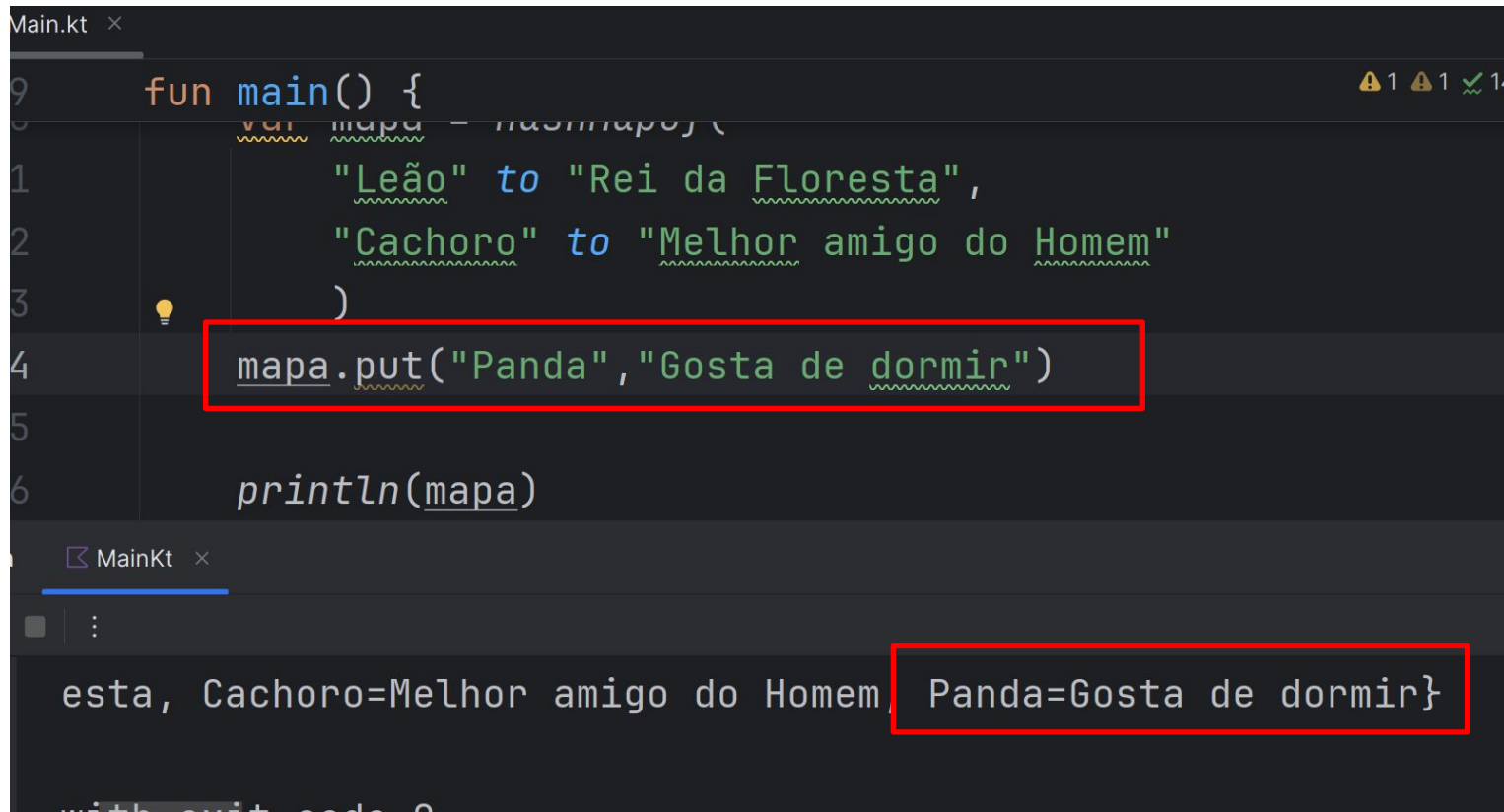


```
9 fun main() {  
10     var mapa = hashMapOf(  
11         "Leão" to "Rei da Floresta",  
12         "Cachoro" to "Melhor amigo do Homem.."  
13     )  
14  
15     println(mapa)
```

Run MainKt

"C:\Program Files\Microsoft\jdk-17.0.10.7-hotspot\bin\java.exe" "
{Leão=Rei da Floresta, Cachoro=Melhor amigo do Homem..}

Acrescentando no map..



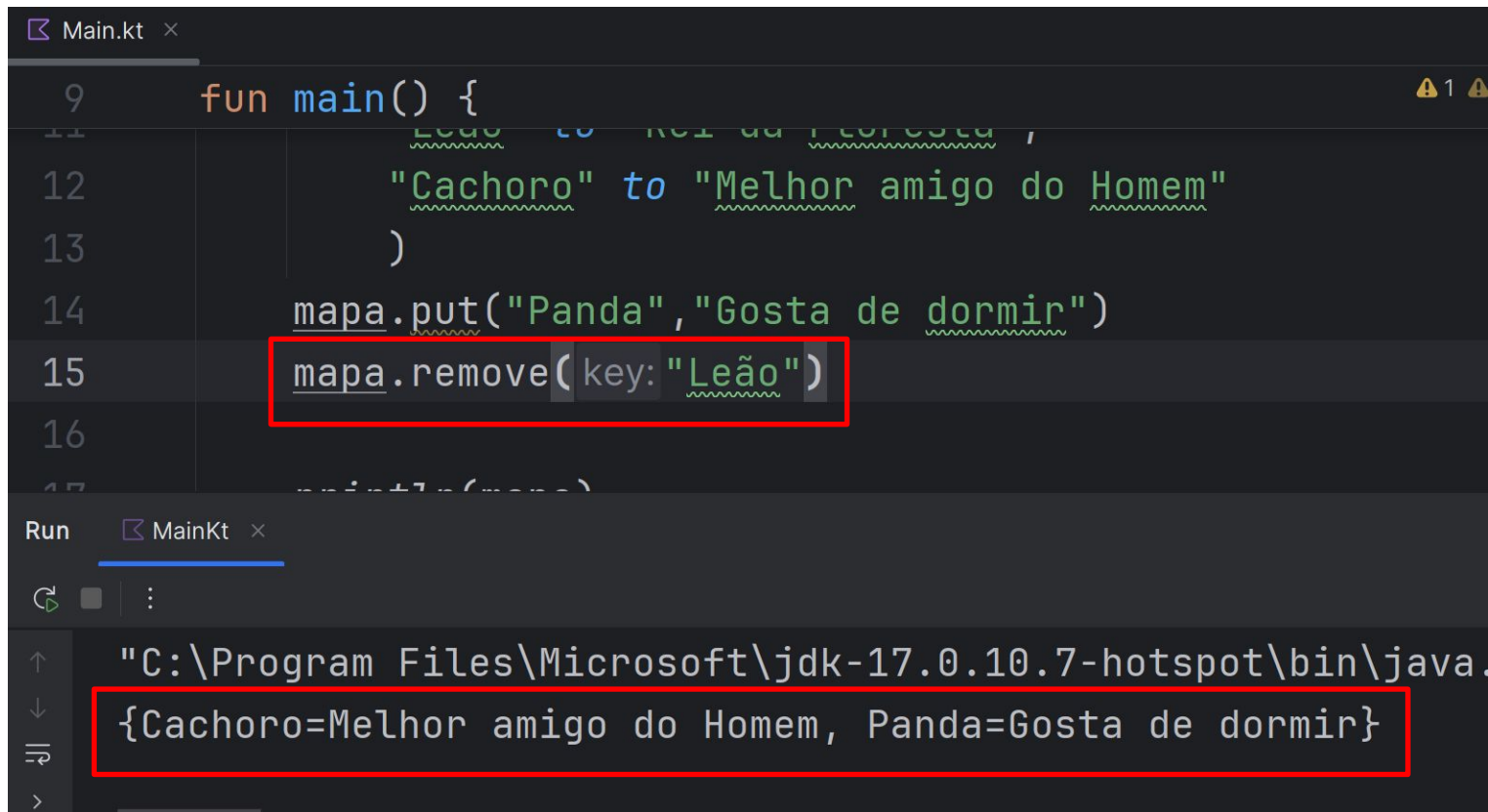
The screenshot shows an IDE with a Kotlin file named `Main.kt`. The code defines a `main` function that creates a map with two entries: `"Leão" to "Rei da Floresta"` and `"Cachoro" to "Melhor amigo do Homem"`. A third entry, `mapa.put("Panda", "Gosta de dormir")`, is added on line 4 and is highlighted with a red box. The map is then printed using `println(mapa)`. The output window at the bottom shows the resulting map: `{Leão=Rei da Floresta, Cachoro=Melhor amigo do Homem, Panda=Gosta de dormir}`, with the new entry highlighted by a red box.

```
9 fun main() {  
10     val mapa = HashMap<>()  
11     mapa.put("Leão", "Rei da Floresta",  
12     "Cachoro" to "Melhor amigo do Homem"  
13 )  
14     mapa.put("Panda", "Gosta de dormir")  
15  
16     println(mapa)  
17 }
```

MainKt x

esta, Cachoro=Melhor amigo do Homem, Panda=Gosta de dormir}

Removendo no map..



```
9 fun main() {  
11     "Cachoro" to "Melhor amigo do Homem"  
12 )  
13  
14     mapa.put("Panda", "Gosta de dormir")  
15     mapa.remove(key: "Leão")  
16  
17     print(mapa)
```

Run MainKt

"C:\Program Files\Microsoft\jdk-17.0.10.7-hotspot\bin\java.
{Cachoro=Melhor amigo do Homem, Panda=Gosta de dormir}

Enum

Definição de Enum..

```
/*  
💡 Enum(enumeração) -> É um tipo de dados que consiste  
em um conjunto de constantes.  
*/
```

Exemplo do uso do Enum..

```
enum class StatusPedido{  
    APROVADO, REPROVADO, PROCESSANDO  
}  
  
class Pedido{  
    var status:StatusPedido = StatusPedido.PROCESSANDO  
}
```

Exemplo do uso do Enum..

```
✓ fun main() {  
    val pedido = Pedido()  
  
    if(pedido.status == StatusPedido.PROCESSANDO){  
        println("Pedido está sendo processado..")  
    }  
}
```

Dúvidas?