

Mobile Application Development

Prof. Fernando Pinéo

Introdução a linguagem de programação Kotlin Parte II

Modificadores de acesso

Modificadores de acesso

- **private** -> Visível apenas dentro da classe
- **protected** -> Funciona como o "private", mas é visível em subclasses
- **public** -> Visível em todo lugar, caso não seja definido é o padrão..

Exemplo..

```
build.gradle.kts (ModificadoresDeAcesso)  Main.kt x
2
3  @↓  open class Animal{
4      var nome = "Potato"
5
6      fun dormir(){
7          println("Dormir...")
8      }
9  }
10
11  class Cachorro :Animal(){
12      fun exibirNome(){
13          println("Exibindo nome...$nome")
14      }
15  }
16
17  ▶ fun main() {
18      val animal = Animal()
19      println(animal.nome)
20  }
```

Definindo como ***private***, propriedade e função

FIAP

The screenshot shows an IDE with two tabs: 'build.gradle.kts (ModificadoresDeAcesso)' and 'Main.kt'. The 'Main.kt' tab is active, displaying the following Kotlin code:

```
1 package org.example
2
3 open class Animal{
4     private var nome = "Potato"
5
6     private fun dormir(){
7         println("Dormir...")
8     }
9 }
10
11 class Cachorro :Animal(){
12     fun exibirNome() {
```

The words 'private' on lines 4 and 6 are highlighted with red boxes. Below the code editor, the 'Build' panel shows a 'Build ModificadoresDeAcesso: failed' message with 2 errors. The error details for 'Main.kt' are shown, with the message 'Cannot access 'var nome: String': it is private in 'org/example/Animal'. :19' highlighted by a red box.

Build Sync x Build Output x

❗ Build ModificadoresDeAcesso: failed At 18/08/2024 21:25 with 2 errors 809 ms

❗ :compileKotlin 1 error 369 ms

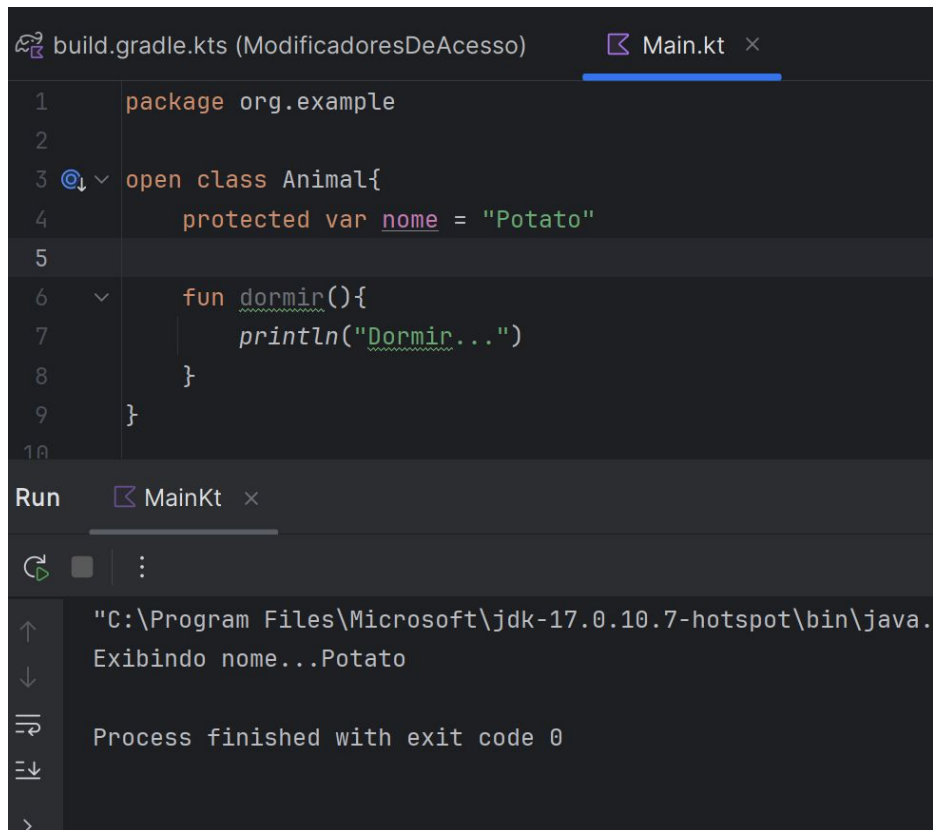
❗ Main.kt src\main\kotlin 1 error

❗ Cannot access 'var nome: String': it is private in 'org/example/Animal'. :19

❗ Compilation error

No slide anterior, ocorreu um erro devido o atributo e método estarem como ***private...***

Exemplo do Protected



The screenshot shows an IDE with two tabs: 'build.gradle.kts (ModificadoresDeAcesso)' and 'Main.kt'. The 'Main.kt' tab is active, displaying the following Kotlin code:

```
1 package org.example
2
3 @v open class Animal{
4     protected var nome = "Potato"
5
6     fun dormir(){
7         println("Dormir...")
8     }
9 }
10
```

Below the code editor, the 'Run' tab is active, showing the execution output:

```
Run MainKt x
C:\Program Files\Microsoft\jdk-17.0.10.7-hotspot\bin\java.
Exibindo nome...Potato
Process finished with exit code 0
```


Protected

- Um pouco parecido com o **private**, porém ao indicar atributo ou método com **protected**, as classes filhos/filhas possuem acesso.
- Por este motivo retornou informações no slide anterior.

Operadores Aritméticos – Básicos

- Somar (+)
- Subtrair (-)
- Multiplicar (*)
- Divisão (/)
- Módulo (%) – Resto de uma divisão

Operadores Relacionais e Lógicos

- *Operadores Relacionais*

- == (Igual a)
- != (Diferente)
- > (Maior que)
- < (Menor que)
- >= (Maior ou igual)
- <= (Menor ou igual)

- *Operadores Lógicos*

- && (e)
- || (ou)

O que retorna do teste a seguir?

```
fun main() {  
    println(5==5)  
}
```

Controle de fluxo – IF e ELSE

```
fun main() {  
    if(true){  
        println("Verdadeiro...")  
    }  
}
```

Exemplo, aninhamento

```
▶ fun main() {  
    var idade = 18  
    if(idade < 14){  
        println("Criança...")  
    } else if(idade >= 14 && idade < 18){  
        println("Adolescente...")  
    } else{  
        println("Adulto..")  
    }  
}
```

Controle de fluxo - When

- Lembra a estrutura Switch desenvolvidas em java..

Exemplo do uso When..

```
fun main() {  
    var opcao = 3  
  
    when(opcao){  
        1 -> {  
            println("Opção 1 selecionada")  
        }  
        2 -> println("Opção 2 selecionada")  
        3 -> println("Opção 3 selecionada")  
        else -> println("Nenhuma opção selecionada")  
    }  
}
```


Opções combinadas

```
fun main() {  
    var opcao = 3  
  
    when(opcao){  
        1,2 -> {  
            println("Opção 1 selecionada")  
        }  
        3 -> println("Opção 2 selecionada")  
        4 -> println("Opção 3 selecionada")  
        else -> println("Nenhuma opção selecionada")  
    }  
}
```

O usuário pode escolher opção número 1 ou 2 neste exemplo..

Null Safe

- Segurança que o Kotlin oferece na hora de você escrever seus códigos..

```
fun main() {  
    var nome : String? = null  
    println(nome)  
}
```


MainKt x

⋮


"C:\Program Files\Microsoft\jdk-17.0.10.
null

Loops for e while


- While

```
fun main() {  
    var numero = 1  
    while (numero<5){  
        println("Executando..")  
        //Incremento  
         numero++  
    }  
}
```

For

```
fun main() {  
    for(numero in 1 ≤ .. ≤ 5){  
         println("Executando...")  
    }  
}
```

For mostrando o índice

```
fun main() {  
    val nomes = arrayOf("Fernando", "Elisabette", "Pedro")  
  
    for((indice, nome) in nomes.withIndex()){  
         println("$indice - $nome")  
    }  
}
```

Dúvidas?