

Padrões de projeto e de código

Este documento foi criado com o intuito de documentar as convenções de código utilizadas durante o curso. Uma vez que a linguagem Kotlin faz a utilização da JVM, vamos utilizar convenções que são usadas no Java. As convenções são descritas no site oficial da linguagem Kotlin [1].

Idioma

O idioma utilizado no curso será inglês, uma vez que os métodos por padrão compartilhados em repositórios e também em documentações oficiais da linguagem são inglês. Dessa maneira, não teremos métodos em diferentes linguagens.

Os comentários serão português para compreensão mais clara já que se trata de um curso destinado para o Brasil.

Linguagem Kotlin

Convenções de nomes para atributos

- Uso de camelCase para variáveis (Evite underline)
- Tipos começam com letra maiúscula (Int, String, Double).
- Métodos e propriedades começam com letra minúscula.
- Identação com espaços com o auxílio da própria IntelliJ IDEA.

```
val courseName: String = "Curso Kotlin"

fun main(args: Array<String>) {
    val isAwesome = true
    isThisCourseAwesome(isAwesome)
}

// Verifica se o curso é demais!
fun isThisCourseAwesome(isAwesome: Boolean){
    if (isAwesome) {
        println("Com certeza! Estamos falando do $courseName")
    }
}
```

Convenções de Classes

Nome de classes

Classes começam sempre com letra maiúscula. Classes com poucos parâmetros podem ser escritas em uma única linha.

```
// Classe com poucos parâmetros  
class Person(val name: String, val yearOfBirth: Int)
```

Classes com construtores mais longos devem ser formatadas para que cada propriedade fique em uma linha. Além disso, o parênteses para fechar o construtor deve ficar em uma nova linha.

```
// Classe com vários parâmetros  
class Course(  
    val title: String,  
    val description: String,  
    val students: Int,  
    val rate: Double  
)
```

Ponto e vírgula

A linguagem deixa como opcional o uso do ponto e vírgula e dessa maneira, como Kotlin preza pela limpeza do código, o uso do ponto e vírgula é desencorajado. Caso seja declarado na IDE, não haverá problema de compilação, mas haverá o aviso constante de que não é necessário.

Unit (Void)

O tipo Unit corresponde a nulo. Em funções que não há retorno, não há necessidade de declarar o retorno como Unit, basta deixar em branco. Caso a função possua retorno, então declaramos qual o seu tipo.

```
fun noReturn() {  
    // ...  
}  
  
fun sayMyName(name: String) : String {  
    return "Hey $name"  
}
```

Referências

[1] <https://kotlinlang.org/docs/reference/coding-conventions.html>, disponível em 21/06/2017.