Object expressions

Object expression é o jeito que **Kotlin** cria o**bjetos anônimos**. **Declarando a variável com object = { } dá pra passar propriedades e funções sem precisar de uma classe explicita referenciando o tipo do Objeto:**

```
val fran = object {
    val nome : String = "Franciane"
    val cpf : String = "123.456.278-1"
    val senha : String = "lalael"

    fun autentica(senha : String) = senha == this.senha
}
```

A **maneira mais comum de usar esse tipo de Sintaxe** é quando precisamos fazer o uso de alguma implementação, mas **não o bastante para que isso vire uma classe na aplicação**. **Objetos anônimos** são usados em escopo de função *(executa a função e morre o objeto).*

Por exemplo, eu **preciso de uma implementação pra um trecho específico do código**, então eu crio um **objeto anônimo e faço essa implementação nesse determinado trecho** e já era *(muito parecido com classes anônimas do Java).*

Object Declaration

Object declaration é um jeito do **Kotlin** criar **singletons *(um objeto único que vai fazer parte do programa do início ao fim dele).*** A sintaxe é parecida com a do ***Object expression***, mas o ***delcaration* em sentido** é meio que o oposto, ele não pode ficar em escopo de função, visto que ele é um singleton.

Então um **object declaration como propriedade de uma classe representa valores que se aplicam independente de instância**, é um **objeto singleton da CLASSE *(É como se fosse um atributo static de um objeto Java):***

```
abstract class Conta (
    val titular :Cliente,
    val numeroConta: String)
{
    var saldo = BigDecimal( val: 0.0)
    private set

    object Contador {
        var total: Int = 0
         private set
    }
```

Sendo **um objeto eu posso mudar a visibilidade das propriedades dele para PRIVADO**, mas se eu quisesse que minha **classe** alterasse esse **objeto declarado** *(que está no escopo dela)* eu não conseguiria. **A não ser que eu use um recurso do Kotlin.**

Esse recurso se chama **companion Object,** quando um **objeto declarado** está no **escopo de uma classe** ele pode ser declarado como um **companion**, dessa maneira as **propriedades dele ficam acessíveis a classe à qual ele faz parte:**