AULA 8

Refatorando projetos





SUPER TRUNFO

Super trunfo é um jogo de cartas. Veja o código do aplicativo Super trunfo para entender como o jogo funciona.

https://github.com/marcellalcs/supertrunfo

O QUE É PROGRAMAÇÃO ORIENTADA À OBJETOS

Objeto: coisa material ou abstrata que pode ser percebida pelos sentidos e descrita por meio das suas **características**, **comportamento** e **estado** atual.

QUAIS OBJETOS EXISTEM NO APP SUPER TRUNFO?

- JOGADOR
- VEÍCULO
- CONDUTOR
- CARTAS
- ...

CLASS - KOTLIN

É um modelo de código de programa extensível para criar objetos, fornecendo valores iniciais para o estado (variáveis de membro) e implementações de comportamento (funções ou métodos de membro)

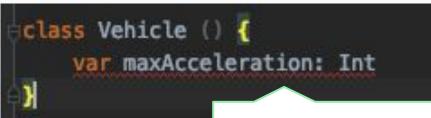
VEHICLE CLASS - KOTLIN



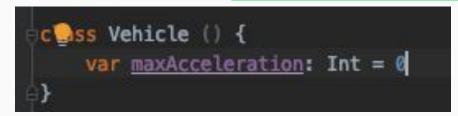
Para criar uma instância

val newVehicleOne = Vehicle()

VEHICLE CLASS - KOTLIN



Para definir um atributo desta forma, precisamos inicializá-lo



VEHICLE CLASS - KOTLIN

```
val newVehicleOne = Vehicle()
newVehicleOne.maxAcceleration = 100
```

Getters e setters são definidos automaticamente para qualquer propriedade

```
class Vehicle () {
    var maxAcceleration: Int = 0
        get() = field * 3
        set(value) {
        field = value/3
    }

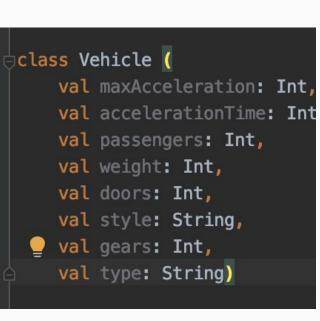
É possível sobre escrever os getters e setters, adicionando comportamentos personalizados
```

DESAFIO

```
val newVehicleOne = Vehicle()
newVehicleOne.maxAcceleration = 100
val maxAcceleration : Int = newVehicleOne.maxAcceleration
```

Qual o valor de maxAcceleration?

Podemos trabalhar com as propriedades dos objetos declarando-as como no exemplo anterior. Mas essa **não é a forma mais comum de declarar propriedades de classes em Kotlin.**



```
val newVehicleOne = Vehicle(
     maxAcceleration: 100,
     accelerationTime: 120,
     passengers: 5,
     weight: 120,
     doors: 2,
     style: "sedã",
     gears: 5,
     type: "car")
```

val testAcceleration : Int = newVehicleOne.accelerationTime
newVehicleOne.maxAcceleration = 10

Por que o erro?

DATA CLASS - KOTLIN

```
dataclass.kt

fun main(args: Array<String>) {
            val vehicleOne = Vehicle( name: "Uno",
                                                    accelerationTime: 10)
            val vehicleTwo = Vehicle( name: "Uno",
                                                    accelerationTime: 10)
            if(vehicleOne == vehicleTwo){
                print("mesmo carro")
            } else{
                print("carros diferentes")
13
        class Vehicle (var name: String, var accelerationTime: Int)
```

O que será impresso no console?

DATA CLASS - KOTLIN

A linguagem Kotlin introduz o conceito de classes de dados, que representam classes simples usadas como contêineres de dados e não encapsulam nenhuma lógica adicional.

DESAFIO - CLASS DRIVER

Crie a class Driver

ISIBILIDADE

- **Public**: Se você não especificar nenhum modificador de visibilidade, public é usado por padrão, o que significa que suas declarações estarão visíveis em qualquer lugar;
- **Private**: Se você marcar uma declaração como privada, ela só ficará visível dentro do arquivo que contém a declaração;
- **Internal:** Se você marcá-lo interno, ele ficará visível em qualquer lugar no mesmo módulo;
- **Protected:** protegido não está disponível para declarações de nível superior.