LUCAS HONDA TONINI

Lista de Exercícios com Classe, função e métodos em Kotlin

Exercício 01

Criar uma função de extensão da classe String com o seguinte nome “firstName(): String” que irá retornar uma String com o primeiro nome de uma pessoa.

Resposta:

```
fun String.firstName(): String {
    return this.trim().split(" ").first()
}
fun main() {
    val name = "Lucas Honda Tonini";
    print(name.firstName());
}
```

```
Lucas
```

Exercício 02

Criar uma função de extensão da classe String com o seguinte nome “lastName(): String” que irá retornar uma String com o último nome de uma pessoa.

Resposta:

```
fun String.lastName(): String {
    return this.trim().split(" ").last()
}
fun main() {
    val name = "Lucas Honda Tonini";
    print(name.lastName());
}
```

```
Tonini
```

Exercício 03

Crie uma classe temperatura, com base em uma temperatura em graus Celsius, a converta e exiba em Kelvin (K)e Fahrenheit (F), seguindo as fórmulas: F = C * 1.8 +32; K = C + 273,15;

Resposta:

```
class Temperatura(val celsius: Double) {
    fun toFahrenheit(): Double = celsius * 1.8 + 32
    fun toKelvin(): Double = celsius + 273.15

    fun exibir() {
        println("$celsius°C = ${"%.2f".format(toFahrenheit())}°F = ${"%.2f".format(toKelvin())}K")
    }
}
fun main() {
    val t = Temperatura(30.0);
    t.exibir();
}
```

```
30.0°C = 86.00°F = 303.15K
```

Exercício 04

Tem-se um conjunto de dados contendo a altura e o sexo (masculino, feminino) de 10 pessoas. Criar uma classe para calcular e escrever: a) a maior e a menor altura do grupo; b) média de altura dos homens; c) o número de mulheres.

Resposta :

```
class CalculadoraAlturas(val dados: List<Pair<Double, String>>) {

    fun maiorAltura(): Double {
        var maior = Double.MIN_VALUE
        for ((altura, _) in dados) {
            if (altura > maior) {
                maior = altura
            }
        }
        return if (maior == Double.MIN_VALUE) 0.0 else maior
    }

    fun menorAltura(): Double {
```

```
Maior altura: 1.80 m
Menor altura: 1.60 m
Média de altura dos homens: 1.76 m
Número de mulheres: 2
```

```
    fun menorAltura(): Double {
        var menor = Double.MAX_VALUE
        for ((altura, _) in dados) {
            if (altura < menor) {
                menor = altura
            }
        }
        return if (menor == Double.MAX_VALUE) 0.0 else menor
    }

    fun mediaAlturaHomens(): Double {
        var soma = 0.0
        var total = 0
```

```
Maior altura: 1.80 m
Menor altura: 1.60 m
Média de altura dos homens: 1.76 m
Número de mulheres: 2
```

```
    fun mediaAlturaHomens(): Double {
        var soma = 0.0
        var total = 0

        for ((altura, sexo) in dados) {
            if (sexo.equals("masculino", ignoreCase = true)) {
                soma += altura
                total++
            }
        }

        return if (total > 0) soma / total else 0.0
    }
```

```
Maior altura: 1.80 m
Menor altura: 1.60 m
Média de altura dos homens: 1.76 m
Número de mulheres: 2
```

```
    fun numeroMulheres(): Int {
        var contador = 0

        for ((_, sexo) in dados) {
            if (sexo.equals("feminino", ignoreCase = true)) {
                contador++
            }
        }

        return contador
    }
}
fun main() {
    val dados = listOf(
```

```
fun main() {
    val dados = listOf(
        1.75 to "masculino",
        1.68 to "feminino",
        1.80 to "masculino",
        1.60 to "feminino",
        1.72 to "masculino"
    )

    val calc = CalculadoraAlturas(dados)

    println("Maior altura: ${"%.2f".format(calc.maiorAltura())} m")
    println("Menor altura: ${"%.2f".format(calc.menorAltura())} m")
    println("Média de altura dos homens: ${"%.2f".format(calc.mediaAlturaHomens())} m")
    println("Número de mulheres: ${calc.numeroMulheres()}")
}
```

~~**Exercício 05**~~

~~Criar as classes em kotlin de acordo com a modelagem UML abaixo. Exibir os métodos e os atributos de cada classe.~~

CONFORME INFORMADO EM SALA DE AULA, NÃO EXECUTADO

Exercício 06

Criar as classes em kotlin de acordo com a modelagem UML abaixo. Exibir os métodos e os atributos de cada classe.

```kotlin
open class Pessoa(
    var nome: String,
    var endereco: String,
    var telefone: String,
    var bairro: String,
    var cep: Long,
    var cidade: String,
    var estado: String
)

class PessoaFisica(
    var cpf: Long,
    var rg: String
) : Pessoa {}

class PessoaJuridica(
    var cnpj: Long
) : Pessoa {}
```

Exercício 07

Criar as classes em kotlin de acordo com a modelagem UML abaixo. Exibir os métodos e os atributos de cada classe.

```
open class Carro(val cor: String, val modelo: String)

class CarroAnfibio(cor: String, modelo: String) : Carro(cor, modelo) {
    fun acelerarMar() {
        println("Acelerando no mar")
    }
}

class CarroTerrestre(cor: String, modelo: String) : Carro(cor, modelo)

class TestarCarro {
    fun executarTeste() {
        val anfibio = CarroAnfibio("Azul", "Kombi")
```

```
Carro Anfíbio:
Cor: Azul
Modelo: Kombi
Acelerando no mar
Carro Terrestre:
Cor: Vermelho
Modelo: Fusca
Executando teste da classe TestarCarro:
Acelerando no mar
```

```
    fun executarTeste() {
        val anfibio = CarroAnfibio("Azul", "Kombi")
        val terrestre = CarroTerrestre("Vermelho", "Fusca")

        anfibio.acelerarMar()
    }
}
fun main() {
    val anfibio = CarroAnfibio("Azul", "Kombi")
    val terrestre = CarroTerrestre("Vermelho", "Fusca")

    println("Carro Anfíbio:")
    println("Cor: ${anfibio.cor}")
    println("Modelo: ${anfibio.modelo}")
    anfibio.acelerarMar()

    println("Carro Terrestre:")
    println("Cor: ${terrestre.cor}")
    println("Modelo: ${terrestre.modelo}")

    val tester = TestarCarro()
    println("Executando teste da classe TestarCarro:")
    tester.executarTeste()
}
```