Teste POO 31/03/2025

Nome:_____

Nº Mecanográfico:_____

1) Considere as entidades abaixo:

Avião, Automóvel, Bicicleta, Barco

Crie uma classe para cada entidade, com 3 atributos no mínimo. Sobrescreva o método *toString* para apresentar as informações sobre a classe. Crie um programa em Kotlin que crie 2 objetos para cada classe criada e apresente na tela todos eles.

Considerando a classe abaixo para as questões **2)** e **3)**:

```
class ContaBancaria(val id: String, saldo: Double) {
    private var saldo: Double = if (saldo > 0) saldo else 0.0
}
```

2) Crie as funções de retirada e depósito. Adicione uma exceção para o caso de uma retirada que ultrapasse o valor do saldo. Trate essa exceção em um programa apresentando uma mensagem de "Saldo insuficiente."

3) Crie uma lista de 5 contas. Apresente uma entrada para o utilizador ao final para entrar com o código de uma conta e acrescentar ou retirar um determinado valor. Apresente o valor dessa conta ao final da operação.

4) Crie uma interface *Bateria* com uma função *fornecer* e uma outra *carregar*. A função fornecer retir energia da bateria, e a função carregar recebe um argumento para adicionar energia (em watts-Hora - Wh). Crie duas classes, *BateriaRelogio* e *BateriaCarro*, que implementam essa interface. Crie 20 classes de cada tipo para cada uma carregue a bateria e forneça energia com valores aleatórios. A energia não pode ficar abaixo de 0. Use exceções se necessário.

5) Escreva uma classe abstrata chamada *Item*. Um *Item* deve conter:

- Um nome
- Um tipo
- Um preco (não pode ser menor que 0)

Escreva as classes *Culinaria*, *Automotivo* e *Decoracao*, subclasses de *Item*

- *Culinaria* contém um campo que indica a validade em dias (Int)
- *Automotivo* contém um campo que indica o modelo (String)
- *Decoracao* contém um campo que indica o compartimento (String)

Crie um programa com uma lista com 3 de cada um desses itens, e crie duas funções

- dado uma lista de itens retorne o item com o menor preco dessa lista
- dado uma lista de itens culinários retorne o de maior validade

Teste as funções com os itens criados.

Alguns computadores da escola aceitam somente a opção de baixo no caso de entrada do utilizador:

```
readln().toDouble()
readLine()!!.toDouble()
```

Para usar aleatórios:

```
import kotlin.random.Random

Random.nextInt(final)
Random.nextInt(comeco, final)
Random.nextDouble() // entre 0 e 1
Random.nextDouble(finalDouble)
Random.nextDouble(comecoDouble, finalDouble)
```