

Aluno: Devair Dener Darolt

UDESC - Universidade do Estado de Santa Catarina
Professor: Vinícius Takeo
Curso de Desenvolvimento Android com Flutter

vtkwki@gmail.com
github.com/takeofriedrich
Módulo I - Básico

Os exercícios devem ser entregues até o dia **08/05/2022**, às 23h59, no Classroom, os arquivos devem ser compactados em um arquivo *.zip* ou *.tar*.

Aulas 1 e 2: Dart Estruturado e Orientado a Objetos

1. Instale o **Dart** e o **Flutter** e execute no Terminal ou no Prompt de Comando ou PowerShell os comandos: “**dart --version**” e “**flutter --version**”. Tire uma *screenshot* com os resultados exibidos.

```
PS C:\Windows\system32> dart --version
Dart SDK version: 2.16.2 (stable) (Tue Mar 22 13:15:13 2022 +0100) on "windows_x64"
PS C:\Windows\system32> flutter --version
Flutter 2.10.5 • channel stable • https://github.com/flutter/flutter.git
Framework • revision 5464c5bac7 (3 weeks ago) • 2022-04-18 09:55:37 -0700
Engine • revision 57d3bac3dd
Tools • Dart 2.16.2 • DevTools 2.9.2
```

2. Implemente uma função em Dart que receba uma lista de inteiros **obrigatória** e um parâmetro **opcional** booleano chamado “**sucessor**”. Defina o valor de “sucessor” como sendo false por default. A função deve retornar a **soma dos elementos** caso o valor de “sucessor” for falso e a **soma dos sucessores de cada elemento** da lista caso o valor de “sucessor” seja true.

entrada		saída
lista	sucessor	
0,1,2,3,4	true	15
0,1,2,3,4	false	10

```
exercício2.dart > ...
Run | Debug
1 void main(List<String> args) {
2   List<int> lista = [1, 2, 3, 4, 5];
3   somarElementos(lista);
4   somarElementos(lista, true);
5 }
6
7 void somarElementos(List<int> lista, [bool sucessor = false]) {
8   int soma = 0;
9   for (int i = 0; i < lista.length - 1; i++) {
10    soma += lista[i];
11  }
12  if (sucessor) {
13    soma += lista[lista.length - 1];
14  }
15  print(soma);
16 }
```

3. Copie e modifique o item anterior convertendo os parâmetros de entrada da função anterior como **parâmetros nomeados**.

```
exercicio3.dart > ...
Run | Debug
1 void main(List<String> args) {
2   List<int> lista = [1, 2, 3, 4, 5];
3   somarElementos(lista: lista);
4   somarElementos(lista: lista, sucessor: true);
5 }
6
7 void somarElementos({required List<int> lista, bool sucessor = false}) {
8   int soma = 0;
9   for (int i = 0; i < lista.length - 1; i++) {
10    soma += lista[i];
11  }
12  if (sucessor) {
13    soma += lista[lista.length - 1];
14  }
15  print(soma);
16 }
```

4. Implemente uma classe em Dart que represente um **Aluno**. Cada aluno possui um nome, idade e uma lista de notas. A classe deve possuir um **método construtor** que recebe todos os atributos além de um **construtor vazio**. Implemente um método chamado **calcularMedia()**, que retorna a média das notas do aluno. Implemente também um método **isAprovado()** que retorna true caso a média seja maior que 7 e false caso contrário.

R: <https://github.com/devairdarolt/Flutter/tree/master/Modulo1/exercicio4>

```
PROBLEMS 5 DEBUG CONSOLE TERMINAL GITLENS TIMELINE OUTPUT
Nome: João,
Idade: 20,
Notas: [7.997855763811163, 5.094374225748282],
Média: 6.5461149947757225,
Situação: Reprovado

Nome: Maria,
Idade: 18,
Notas: [0.4327331261953449, 1.4684954436193052, 4.009233164288255, 9.906833708305578, 6.543882661351477, 5.924791724707035, 5.508092411557094, 5.391750131360439],
Média: 4.897126546423066,
Situação: Reprovado
```

5. Estenda a classe Aluno em duas: **AlunoMatutino** e **AlunoVespertino** sobrescrevendo o método **isAprovado()** onde os alunos do período matutino são aprovados apenas com média maior ou igual a 7, enquanto que os alunos do período vespertino são aprovados apenas com média maior ou igual a 6.

R: <https://github.com/devairdarolt/Flutter/tree/master/Modulo1/exercicio5>

```
exercicios > main.dart > fx main
1 import 'dart:math';
2
3 import 'models/aluno.dart';
4 import 'models/aluno_matutino.dart';
5 import 'models/aluno_espertino.dart';
6
7 Run | Debug
8 void main(List<String> args) {
9   AlunoMatutino alunoMatutino = AlunoMatutino();
10  alunoMatutino.nome = 'Tonho';
11  alunoMatutino.idade = 28;
12  alunoMatutino.notas = [];
13  geraNotas(alunoMatutino, 3);
14  print(alunoMatutino.toString());
15
16  AlunoVespertino alunoVespertino = AlunoVespertino();
17  alunoVespertino.nome = 'Joana';
18  alunoVespertino.idade = 25;
19  alunoVespertino.notas = [];
20  geraNotas(alunoVespertino);
21  print(alunoVespertino.toString());
22
23 void geraNotas(Aluno aluno, [int nrProvas = 2]) {
24   for (int i = 0; i < nrProvas; i++) {
25     double nota = Random().nextDouble() * 10;
26     aluno.addNota(nota);
27   }
28 }
29
```

```
PROBLEMS 5 DEBUG CONSOLE TERMINAL GITLENS TIMELINE OUTPUT
Nome: Tonho,
Idade: 28,
Notas: [7.877771633276463, 9.21933618090544, 2.475102104624317],
Média: 6.524069972935407,
Situação: Reprovado

Nome: Joana,
Idade: 25,
Notas: [6.6245783015191515, 5.026368637847759],
Média: 5.825473469683455,
Situação: Reprovado

Exited
```

6. Instancie dois objetos de cada uma das classes (Aluno, AlunoMatutino e AlunoVespertino) e defina valores para nome, idade e lista de notas, apresentando no terminal/console o resultado da chamada do método **isAprovado()**;

```
PROBLEMS 5 DEBUG CONSOLE TERMINAL GITLENS TIMELINE OUTPUT
Nome: Tonho,
Idade: 28,
Notas: [7.877771633276463, 9.21933618090544, 2.475102104624317],
Média: 6.524069972935407,
Situação: Reprovado

Nome: Joana,
Idade: 25,
Notas: [6.6245783015191515, 5.026368637847759],
Média: 5.825473469683455,
Situação: Reprovado

Exited
```