

By Reus/Cury -> 2 semanas

By Giovane -> 3 semanas

By Welinton -> 1 mês

Todos os exercícios devem ser realizados no dartpad.

Todos os exercícios devem ser salvos em um diretório no github de vcs chamado: aulas-dart

Todos os códigos devem estar em inglês.

Para todas as classes, deverá ser instanciado no mínimo 3 objetos de cada, enviando valores diferente justamente para abranger todos os cenários.

1 – Crie uma interface com apenas um método que receba duas strings e retorne as duas primeiras letras da primeira string concatenadas com a primeira letra da segunda string.

Essa interface deve ser implementada em uma classe concreta, implementando a seguinte funcionalidade:

Sempre que o usuário digitar nome e sobrenome deve retornar apenas 3 letras em maiúsculo, sendo:

A primeira e segunda letra da primeira string (nome)

A primeira letra da segunda string (sobrenome)

Exemplo:

Entrada: Thiago Cury

Saída do método: THC

2 – Crie uma classe contendo um método que imprima o nome 20 vezes através de um laço.

3 – Crie uma classe contendo um método que retorne uma lista contendo 10 nomes de pessoas fixos.

4 – Crie uma classe contendo um método que retorne um json com uma lista de 10 nomes de animais, como por exemplo:

```
[  
  {"name": "Rex"},  
  {"name": "Flopi"},  
  ...  
]
```

5 – Crie uma classe contendo um método que retorne um json contendo 5 objetos com as seguintes características: id, description, value

6 – Crie uma classe contendo dois métodos, sendo que eles devem receber por parâmetro um valor inteiro:

- O primeiro deve retornar os números de 0 até o valor inteiro recebido de forma crescente;
- O segundo deve retornar os números do valor inteiro recebido até zero de forma DEcrescente;
- Se o valor for negativo, informar o usuário que o valor não é aceito

7 – Crie um mixin para um método que receba uma string e valide se ela contém uma arroba. Vamos imaginar a validação de um email, onde deve e só pode haver uma única arroba. Além da arroba, a string deve conter “.com”, indicando que é um email válido. Esse mixin deverá ser acoplado em uma classe criada por você.

8 – Crie uma classe chamada instituição(IES). Essa classe receberá uma lista de objetos, onde cada objeto terá os parâmetros relacionados a uma instituição. Parâmetros: Nome, alias, id e hostname.
Para fazer o mapeamento deverá ser utilizado um construtor nomeado chamado fromMap, utilizando o design pattern factory.

ps.: Para criar objeto, utilize uma string mockada no main.

9 – Crie um método que receba uma string e retorna a mesma string com letras maiúsculas e minúsculas intercaladas.

Por exemplo:

O método recebe: Thiago

O método deverá retornar: ThlaGo

Esse método deverá ser adicionado com extension a classe String.

10 – **(by Welinton)** Lista de Objetos a serem mapeados

```
[
  {
    id: 1,
    name: Welinton,
    gender: 2
  },
  {
    id: 2,
    name: Julia,
    gender: 1
  },
  {
    id: 3,
    name: Laís,
```

```

        gender: 1
    },
    {
        id: 4,
        name: Raj,
        gender: 2
    },
    {
        id: 5,
        name: Tammy,
        gender: 3
    },
]

```

- Com base na lista acima, faça uma classe Pessoa que receba os parâmetros dos objetos. Para criação dos objetos utilize um método construtor nomeado fromMap. Utilizando o design pattern Factory.

Após a criação da lista de pessoas, crie um mixin para ser implementado apenas pela classe Pessoa, que terá um método que recebe o gender inteiro e retorne o gender como String.

Ex.:

```

gender 1 = Feminino
gender 2 = Masculino
gender 3 = Outros

```

11 – Crie uma classe que receba os atributos necessários para realizar o cálculo de IMC. Essa classe deve obrigatoriamente utilizar métodos assessores(gets/sets) para o recebimento e envio de informações para os atributos. O retorno final deve ser os valores enviados junto com o cálculo de IMC.

12 – Faça uma classe que ao ser instanciada receba um número inteiro e gere automaticamente números randômicos inteiros com base no número recebido.

Após o usuário poderá chamar dois métodos, são eles:

- método para retornar somente os números pares
- método para retornar somente os números ímpares

13 – **(by Reus)** Crie uma classe que receba um json, como por exemplo:

```

[
{"name": "Josias da Silva Filho", "type": 1, "document": "00271750020"},
{"name": "Burguer do Gurgel" , "type": 2, "document": "76688432000107"}
...
]

```

A classe deverá fazer o mapeamento para um objeto pessoa física e/ou jurídica, conforme o type. Se for 1: física, se for 2 pessoa jurídica.

A classe também deverá ter 1 ou 2 métodos (siga os preceitos do clean code) para realizar a validação do CPF e do CNPJ, onde deve ser retornado uma mensagem na tela informando se o cpf/cnpj é válido ou inválido.

Observação: A classe deve receber o documentation como string, no entanto não podemos realizar cálculo com string. Para isso você deve splitar essa string, criar as listas necessárias para guardar os valores, e após, realizar os cálculos de validação específicos para cpf e cnpj.

Para realizar os testes, use os criadores de cpf e cnpj online:

<https://www.geradordecpf.org/>

<https://www.freetool.dev/pt/cnpj-gerador-validador>

Real life exercises

**“Dias fáceis geram profissionais fracos,
dias difíceis geram profissionais fortes”**

ATENÇÃO: Os exercícios a seguir não devem ser postados no github pessoal.

Com base nos códigos, refaça as calculadoras da melhor forma possível, utilizando tudo que você aprendeu de técnica até agora.

14 – (nível easy) [calc49-karnofsky.html](#)

15 - (nível easy) [calc53-frailty.html](#)

16 – (nível medium) Calculadora de déficit de bicabornato: [calc48-deficit_de_bicarbonato.html](#)

17 – (nível medium) Calculadora de colesterol: [calc54-ldl.html](#)

18 – (nível medium) Calculo de peso ideal fórmula de Devine e Lorentz: [calc10-peso_ideal.html](#)

Os exercícios a seguir são exercícios voltados para o teu crescimento profissional.

19 – Pegar um exercício entre o 1 e o 12 e fazer uma postagem no LinkedIn sobre o Dart. A ideia é focar nas técnicas atuais de desenvolvimento que você está aprendendo, utilizando nos exercícios e o quanto isso vai te ajudar em uma visão de futuro.

Essa postagem deverá marcar: Thiago Espíndola Cury, Welinton Prediger, Giovane de Lored, vcs, Andressa Boll e Eduardo Reus.

20 – Tarefa final. Cada um de vocês pegará um código diferente da calculadora do +PSI e fará uma postagem no LinkedIn.

Essa postagem poderá conter textos sobre:

- Explicação sobre o trabalho que foi realizado nas mais de 50 calculadoras, conversão das linguagens HTML+CSS+JS para Dart.
- Comentar o que foi utilizado de técnica no Dart para a conversão da calculadora para a nova linguagem
- Como está sendo a jornada de vcs no programa de estágio da +A Educação
- Falar sobre o treinamento de Dart

Essa postagem deverá marcar: Thiago Espíndola Cury, Welinton Prediger, Giovane de Lored, vcs, Andressa Boll e Eduardo Reus.

Essa postagem deverá ter os nomes das referências que estão na própria calculadora, como por exemplo:

Revisão técnica: Lucas Seiki Mestre Okabayashi

Referência: Child CG, Turcotte JG. Surgery and portal hypertension. In: Child CG. The liver and portal hypertension. 3rd ed. Philadelphia: Saunders; 1964. p. 50-64.

