

Questões de Dart: Funções, Funções de Seta e Parâmetros com Contexto e Melhoria Contínua

Nível Fácil

1. Definição de Classe

- **Contexto e Tarefa:** Um aplicativo de contatos precisa armazenar informações básicas de uma pessoa. Escreva uma classe Pessoa que tenha as propriedades nome e idade.

2. Construtor Padrão

- **Contexto e Tarefa:** Um programa de gerenciamento de alunos precisa inicializar objetos Aluno com o nome e a matrícula. Escreva um construtor padrão para a classe Aluno que receba nome e matricula.

3. Método de Saudação

- **Contexto e Tarefa:** Um sistema de eventos deseja cumprimentar os participantes pelo nome. Adicione um método saudacao na classe Participante que imprima uma saudação personalizada.

4. Propriedades Nomeadas e Construtor

- **Contexto e Tarefa:** Um aplicativo de endereço precisa criar objetos Endereco com propriedades nomeadas rua, cidade e estado. Escreva a classe Endereco com um construtor que inicialize essas propriedades.

5. Método de Formatação

- **Contexto e Tarefa:** Um sistema de CRM precisa formatar os detalhes de contato de um cliente. Adicione um método formatarContato na classe Cliente que retorne uma string com o nome e telefone formatados.

6. Classe com Método Estático

- **Contexto e Tarefa:** Um aplicativo de utilidades precisa verificar se um número é par. Adicione um método estático ePar na classe Util que retorne se um número é par.

7. Classe com Método de Cálculo de Área

- **Contexto e Tarefa:** Um aplicativo de geometria precisa calcular a área de um retângulo. Escreva uma classe Retangulo com propriedades largura e altura, e um método calcularArea que retorne a área do retângulo.

8. Construtor com Valor Padrão

- **Contexto e Tarefa:** Um programa de descontos precisa inicializar objetos Produto com preço original e um desconto opcional. Escreva um construtor para a classe Produto que inicialize essas propriedades, com o desconto tendo um valor padrão de 0.

9. Classe com Método de Exibição de Dados

- **Contexto e Tarefa:** Um sistema de biblioteca precisa exibir informações sobre os livros. Adicione um método `exibirInformacoes` na classe `Livro` que imprima o título e o autor.

10. Método de Cálculo de Perímetro

- **Contexto e Tarefa:** Um aplicativo de geometria precisa calcular o perímetro de um quadrado. Escreva uma classe `Quadrado` com a propriedade `lado` e um método `calcularPerimetro` que retorne o perímetro.

Nível Intermediário

11. Método de Cálculo de Juros Simples

- **Contexto e Tarefa:** Um banco deseja calcular os juros simples sobre um depósito. Adicione um método `calcularJurosSimples` na classe `ContaBancaria` que receba `principal`, `taxa` e `tempo`, e retorne o valor dos juros simples.

12. Método de Verificação de Idade

- **Contexto e Tarefa:** Um sistema de controle de acesso precisa verificar se uma pessoa é maior de idade. Adicione um método `eMaiorDeIdade` na classe `Pessoa` que retorne se a idade é maior ou igual a 18.

13. Classe com Construtor Nomeado

- **Contexto e Tarefa:** Um programa de gestão de produtos precisa criar objetos `Produto` com um construtor nomeado `Produto.desconto` que inicialize a propriedade `preco` com desconto aplicado. Escreva a classe `Produto` com este construtor nomeado.

14. Método de Cálculo de Média

- **Contexto e Tarefa:** Um sistema escolar precisa calcular a média das notas dos alunos. Adicione um método `calcularMedia` na classe `Aluno` que receba uma lista de notas e retorne a média.

15. Método de Ordenação de Lista

- **Contexto e Tarefa:** Um sistema de gestão de eventos precisa ordenar os nomes dos participantes em ordem alfabética. Adicione um método `ordenarNomes` na classe `Evento` que receba uma lista de nomes e retorne a lista ordenada.

16. Método de Conversão de Moeda

- **Contexto e Tarefa:** Um aplicativo de finanças precisa converter valores de dólares para euros. Adicione um método `converterMoeda` na classe `Finanças` que receba um valor em dólares e retorne o valor em euros, considerando uma taxa de conversão fixa.

17. Método de Saudação com Horário

- **Contexto e Tarefa:** Um aplicativo precisa exibir uma saudação ao usuário dependendo do horário do dia. Adicione um método `saudacao`

na classe `Usuário` que receba o nome do usuário e o horário (manhã, tarde, noite) como parâmetros nomeados e imprima uma saudação adequada.

18. Método de Verificação de Palíndromo

- **Contexto e Tarefa:** Um aplicativo de palavras precisa verificar se uma palavra é um palíndromo. Adicione um método `ePalindromo` na classe `Palavra` que receba uma palavra e retorne se ela é um palíndromo.

19. Método de Cálculo de IMC

- **Contexto e Tarefa:** Um aplicativo de saúde precisa calcular o Índice de Massa Corporal (IMC) dos usuários. Adicione um método `calcularIMC` na classe `Saúde` que receba o peso e a altura do usuário como parâmetros e retorne o IMC.

20. Método de Filtragem de Produtos

- **Contexto e Tarefa:** Um sistema de e-commerce precisa filtrar produtos por categoria. Adicione um método `filtrarPorCategoria` na classe `Loja` que receba uma lista de produtos e uma categoria como parâmetros e retorne os produtos que pertencem a essa categoria.

Nível Avançado

21. Método de Busca Binária

- **Contexto e Tarefa:** Um algoritmo de busca precisa encontrar um valor em uma lista ordenada. Adicione um método `buscar` na classe `Busca` que implemente a busca binária e retorne a posição do valor na lista.

22. Método de Cálculo de Potência

- **Contexto e Tarefa:** Um sistema de cálculo precisa determinar a potência de um número. Adicione um método `calcularPotencia` na classe `Matematica` que receba a base e o expoente como parâmetros e retorne a potência.

23. Método de Formatação de Data

- **Contexto e Tarefa:** Um sistema precisa exibir datas em um formato legível. Adicione um método `formatarData` na classe `Data` que receba uma data e retorne a data formatada como "dia/mês/ano".

24. Método de Verificação de Número Primo

- **Contexto e Tarefa:** Um aplicativo matemático precisa verificar se um número é primo. Adicione um método `ePrimo` na classe `Número` que receba um número e retorne se ele é primo.

25. Método de Criação de Lista de Quadrados

- **Contexto e Tarefa:** Um programa precisa gerar uma lista de quadrados dos números de 1 a 10. Adicione um método `gerarListaDeQuadrados` na classe `ListaNumerica` que retorne uma lista contendo os quadrados desses números.

26. Método de Classificação de Idades

- **Contexto e Tarefa:** Um sistema de classificação precisa agrupar pessoas por faixas etárias. Adicione um método `classificarIdades` na classe `Classificacao` que receba uma lista de idades e retorne uma lista de strings com as classificações ("criança", "adolescente", "adulto", "idoso").

27. Método de Conversão de Minutos para Horas

- **Contexto e Tarefa:** Um aplicativo precisa converter um tempo em minutos para horas e minutos. Adicione um método `converterMinutosParaHoras` na classe `Tempo` que receba um número de minutos e retorne uma string no formato "X horas e Y minutos".

28. Método de Geração de Senhas Aleatórias

- **Contexto e Tarefa:** Um sistema de segurança precisa gerar senhas aleatórias para os usuários. Adicione um método `gerarSenhaAleatoria` na classe `Segurança` que gere uma senha aleatória de 8 caracteres contendo letras e números.

29. Método de Cálculo de Média Ponderada

- **Contexto e Tarefa:** Um sistema acadêmico precisa calcular a média ponderada das notas dos alunos. Adicione um método `calcularMediaPonderada` na classe `Academico` que receba uma lista de notas e uma lista de pesos e retorne a média ponderada.

30. Método de Conversão de Temperatura

- **Contexto e Tarefa:** Um aplicativo meteorológico precisa converter temperaturas de Celsius para Fahrenheit. Adicione um método `converterParaFahrenheit` na classe `Temperatura` que receba uma temperatura em Celsius e retorne o valor em Fahrenheit.