

Exercícios

1. Exibição de alunos

Você está desenvolvendo um sistema de cadastro de alunos e precisa criar um programa em Dart para armazenar e exibir suas informações pessoais. Crie um programa que declare e atribua valores às seguintes variáveis:

- nome : deve conter o nome completo do aluno
- idade : deve conter a idade do aluno, em anos
- ehNovato : deve indicar se o aluno entrou no ano atual na escola
- peso : deve conter o peso do aluno, em quilogramas
- notas : deve conter as notas do aluno em 4 avaliações, que podem ir de 0 a 10
- disciplinas : deve conter as disciplinas que o aluno está cursando
- endereco : deve conter o endereço do aluno como texto. No entanto, um aluno pode não ter um endereço cadastrado no sistema ainda.

Após a declaração e atribuição dos valores, exiba na tela as informações do aluno. Certifique-se de exibir o seu nome, a sua idade, se é novato, o seu peso, suas notas, as disciplinas que está cursando e seu endereço, caso possua.

2. Manipulação de redações

Como o sistema está sendo implementado depois das férias escolares, é de praxe que os professores peçam para os alunos escreverem redações sobre o que fizeram

no período.

Para digitalizar o que um aluno escreveu, declare uma variável `redacao` do tipo texto com alguma frase curta, representando a redação de um aluno. Após a declaração da variável, resolva as seguintes situações:

- A redação deve obrigatoriamente falar sobre as férias. Exiba se `redacao` possui em alguma parte o texto “férias”.
- A redação deve ter uma quantidade mínima de palavras. Exiba quantas palavras `redacao` possui. Considere o caractere espaço como delimitador de uma palavra.
- Lendo a redação de alguns alunos, você percebeu que muitos deles estavam escrevendo “mais melhor”! Como está sendo começo de semestre, você deixa passar só dessa vez. Exiba `redacao` com essa correção, substituindo as ocorrências de “mais melhor” por “melhor”.

3. Manipulação de notas

As notas do semestre passado ainda não foram lançadas pois são muitos papéis, e fazer manualmente está demorando muito. Como você é o novo desenvolvedor, você vai precisar manipular as notas de cada aluno. O sistema deve permitir adicionar, remover e ordenar as notas dos alunos, além de extrair uma sublistagem específica.

Utilizando o programa criado na primeira questão, após a declaração das variáveis, resolva as seguintes situações:

- No fim do ano anterior, o professor resolveu dar um presente aos alunos e resolveu dar a nota máxima para todos como uma nota bônus. Adicione a nota 10 a `notas`.

- A prova da terceira avaliação teve muitas questões anuladas por um erro da escola e o diretor decidiu cancelar esta nota. Remova a terceira nota de `notas`.
- O professor precisava saber as notas apenas do primeiro semestre de cada aluno para usar como base posteriormente no Conselho de Classe. Crie uma nova lista chamada `notas1Semestre` contendo apenas as duas primeiras notas de `notas`.
- Alguns alunos, por um engano da secretária da escola, apareceram com uma nota 11 no seu boletim. Remova esta nota de `notas`, caso esteja presente.
- Por fim, ordene `notas` em ordem crescente.

Exiba na tela o resultado das operações realizadas.

4. Manipulação de pesos

Como forma de verificar a saúde dos alunos a cada início de semestre, um nutricionista faz uma medição adicional no peso de cada aluno.

Utilizando o programa criado na primeira questão, após a declaração das variáveis, resolva as seguintes situações:

- A balança do nutricionista dá resultados apenas como texto! Crie uma variável `pesoBalanca`, do tipo `String`, e atribua um valor numérico arbitrário. Crie outra variável `pesoAtual`, do tipo `int`, e transforme o valor de `pesoBalanca` para um valor aceito por `pesoAtual`.
- A balança possui um limite de 70kg, e partir disso ela retorna resultados negativos. Para padronizar todos os alunos, atualize `pesoAtual` para o seu valor absoluto.
- Por fim, arredonde o peso para o valor inteiro mais próximo.

Exiba na tela o resultado das operações realizadas.

5. Mais manipulação de pesos

O programador novo do seu setor achou melhor usar um mapa para organizar alunos e seus pesos, refatorando todo o trabalho que você teve durante todo esse tempo. Cada chave do mapa é o nome de um aluno (`String`) e cada valor é seu peso (`double`).

Utilizando o programa criado na primeira questão, após a declaração das variáveis, crie um mapa chamado `pesos` e resolva as seguintes situações:

- Adicione uma aluna inicial a `pesos` . Seu nome é Carol e ela pesa 58 kg.
- Adicione o aluno atual a `pesos` . Seu nome está armazenado em `nome` e seu peso em `peso` .
- Adicione mais alguns alunos! Defina seu nome e o associe a um peso.
- Armazene o valor correspondente à chave de um aluno que você adicionou em uma variável chamada `pesoNovo` .
- Perdão, a Carol não deveria ser adicionada ainda. Remova a entrada correspondente ao seu nome.
- Exiba se o mapa contém uma chave igual ao seu primeiro nome.

Exiba na tela o resultado das operações realizadas.

Referências

- Dart. [Variables](#): explica variáveis e modificadores em Dart
- Dart. [Null safety codelab](#): ensina nulabilidade em Dart com exemplos práticos
- Dart. [Sound null safety](#): explica a implementação da nulabilidade em Dart
- Dart News & Updates. [Const, Static, Final, Oh my!](#): detalha objetos constantes em Dart e como funcionam na memória

Leituras adicionais

- StackOverflow. [What is the difference between dynamic and Object in dart?](#): explica a diferença entre `dynamic` e `Object?` em Dart
- Amanda Hinchman. [Null Pointer References: The Billion Dollar Mistake](#): explica a origem da preocupação com valores nulos
- Thomas Maximini. [Typescript: Bang operator considered harmful](#): explica o porquê do operador `!` ser considerado [má prática](#) e fornece alternativas ao seu uso; a linguagem é o Typescript, mas a sintaxe também se aplica a Dart