UNIVERSIDADE ESTADUAL DO PIAUÍ – UESPI
CURSO DE BACHARELADO EM COMPUTAÇÃO
DISCIPLINA: PROGRAMAÇÃO II
PROFESSOR: EYDER RIOS

2ª Avaliação

Você está desenvolvendo um programa para gerenciar uma biblioteca digital que contém diferentes tipos de mídia, como audiolivros, filmes e músicas. Cada mídia possui um título e uma duração (em minutos). O programa deve permitir que o usuário (a) adicione novas mídias, (b) liste todas as mídias disponíveis e (c) calcule a duração total de todas as mídias na biblioteca.

Sua tarefa é implementar uma classe abstrata chamada "Media" que represente uma mídia genérica. Essa classe deve possuir os seguintes atributos do tipo “final”:

- Um atributo do tipo enum chamado “type” para armazenar o tipo de mídia (livro, filme ou música).
- Um atributo chamado "title" para armazenar o título da mídia.
- Um atributo chamado "duration" para armazenar a duração da mídia em minutos.

Além disso, a classe deve ter os seguintes métodos:

- Um construtor que recebe o tipo de mídia, o título e a duração como parâmetros e inicializa os atributos correspondentes (todos os atributos devem ser “final”).
- Um *getter* abstrato chamado "name" que retorna o nome da pessoa associada à obra (autor, diretor ou artista).

Em seguida, crie três sub-classes de "Media": "AudioBook", "Movie" e "Music". Cada classe deve adicionar um atributo do tipo “final” específico de cada classe (AudioBook, Movie e Music) e implementar o getter "name" de forma a retorna a informação correspondente.

- A classe "AudioBook" deve adicionar um atributo “final” chamado "author".
- A classe "Movie" deve adicionar um atributo “final” chamado "director".
- A classe "Music" deve adicionar um atributo “final” chamado "artist".

Finalmente, implemente uma classe chamada "DigitalLibrary" que irá armazenar internamento uma lista de mídias e que disponibiliza os seguintes métodos:

- Um método chamado "addMedia()" que permite ao usuário adicionar uma nova mídia à biblioteca. O método deve receber como parâmetro uma instância que seja subclasse de “Media”.
- Um método “loadMedia()” que carrega de um arquivo de dados JSON os dados de mídias.
- Um método chamado "listMedia()" que exibe os detalhes de cada uma das mídias registradas na biblioteca de acordo com o formato especificado no arquivo “output.txt” (ver seção Recursos do Projeto).
  - O método deve possuir um parâmetro posicional opcional *nullable* que indica o tipo de mídia a ser exibida. Se o parâmetro for *null*, todos os tipos de mídia devem ser exibidos, caso contrário apenas as mídias do tipo especificado.
- Um método chamado "totalMediaDuration()" que retorna o tempo total de duração (em minutos) de todas as mídias na biblioteca.
  - O método deve possuir um parâmetro posicional opcional *nullable* que indica o tipo de mídia a ser considerado no cômputo do tempo total de duração. Se o parâmetro for *null*, todos os tipos devem ser considerados no cálculo, caso contrário apenas as mídias do tipo especificado.

REQUISITOS E FUNCIONALIDADES:

1. Ao implementar sua solução utilize de forma eficiente herança e polimorfismo para lidar com diferentes tipos de mídia.
2. Não acrescente nenhum atributo adicional às classes além dos especificados no enunciado.
3. Todos os identificadores utilizados na aplicação devem usar a língua inglesa e observar os princípios de estilo do Effective Dart.
4. A aplicação deverá ser organizada em arquivos e pastas conforme a necessidade.
5. Sua implementação deverá ser executada com o arquivo "main.dart" fornecido (ver seção Recursos do Projeto). Este arquivo não deve ser alterado, com exceção da linha que exibe os nomes dos autores.
6. O projeto deverá ser disponibilizado em um repositório com acesso público do GitHub, com "main" como seu branch principal (default do GitHub). Qualquer outro branch não será considerado.
    - O código fonte do projeto deverá ser armazenado no GitHub (github.com) em um repositório com acesso público.

RECURSOS DO PROJETO:

1. Todos os arquivos necessários para o projeto estão disponíveis no repositório https://github.com/uespi-phb/prog2-20222/tree/main/aval2
2. Os arquivos disponibilizados não deverão ser alterados.

ARQUIVO	DESCRIÇÃO
bin/main.dart	Arquivo principal para teste do código submetido.
lib/utils.dart	Arquivo com funções utilitárias para formatação de saída.
data/media.json	Arquivo contendo dados para execução dos testes.
output.txt	Saída esperada ao executar o programa a partir do arquivo "main.dart"
p2-aval2.pdf	Este arquivo.

COMPOSIÇÃO DA EQUIPE:

1. A implementação poderá ser realizada em equipes de ATÉ 2 (dois) membros;
2. Os autores de cada implementação poderão ser questionados sobre o código implementado, com o objetivo de comprovar a participação de cada membro na execução do projeto;
3. A comprovação de não participação do projeto de algum membro da equipe implicará em:
    - Sua exclusão da equipe, e atribuição de nota zero;
    - Penalidade de 30% na nota da equipe.

INTRUÇÕES PARA REMESSA DO PROJETO:

- Ao finalizar o projeto, remeter o *link* do repositório para o e-mail: eyder@phb.uespi.br com o seguinte assunto: "PROG2 – AVAL2", juntamente com os nomes dos membros da equipe no corpo do e-mail;
- O projeto deverá ser compatível com a versão 2.17.0 (ou superior) do Dart.