

IIª Avaliação – 2023.1- TRABALHO

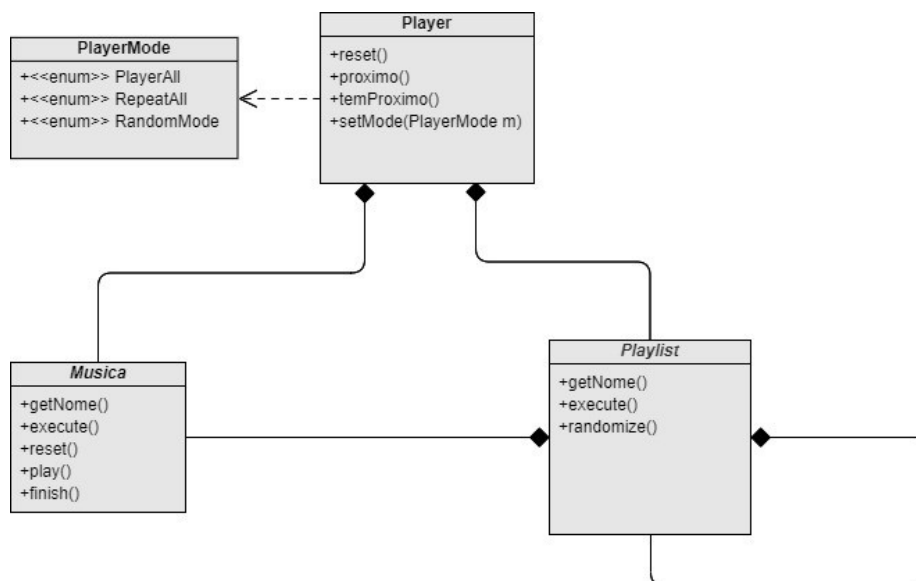
As questões desta avaliação são baseadas no seguinte cenário de classes:

Considere um sistema que mantém uma base de músicas e que se destina a exibição de cifras das músicas, letras e traduções. Os arquivos com letras, cifras, traduções de cada música são mantidos em um diretório e gerenciados por um Resource Loader. As músicas podem ser exibidas apenas com a cifra; ou com a cifra e letra; cifra e tradução; cifra, letra original e tradução; em qualquer combinação possível, inclusive podendo ser ampliada com outras representações.

As playlists são compostas por músicas e outras playlists e (assim como as músicas), podem ser executadas por um Player. A execução consiste na geração da versão string das músicas (apenas cifras, com letras, traduções, etc.), ou no caso das playlists, das músicas que compõem a playlist.

O Player pode operar em três modos: o modo no qual as músicas e playlists são executadas apenas uma vez e na sequência de inserção; o modo em que as músicas e playlists são executadas repetidamente na sequência de inserção (voltando à primeira após a última música), e um terceiro modo de execução das músicas de forma pseudo-aleatória.

Um desenho preliminar das classes componentes do modelo foi indicada pela equipe de projeto e está indicada abaixo:



No diagrama apresentado:

PlayerMode: é uma enumeração que lista os modos de funcionamento do Player

- **PlayerAll:** executa as músicas e playlists uma única vez, respeitando a ordem de inserção.
- **RepeatAll:** executa as músicas e playlists, respeitando a ordem de inserção, mas voltando ao início após a última música ter sido executada.
- **RandomMode:** executa as músicas de forma pseudo-aleatória

Player: é a classe que representa o executor. É uma agregação de músicas e playlists. Seu método `temProximo` e `proximo`, permitem que a aplicação navegue através das músicas e playlists. A navegação é dependente do modo de operação do Player (indicado por um dos PlayerModes)

Música: é a classe que apresenta as notas musicais de uma Música. As notas são apresentadas linha a linha, como apresentado no exemplo a seguir. O método reset coloca na primeira linha da música, o método play traz a linha atual da música e avança o contador de linhas para a próxima linha. O método finish indica se a ainda existem linhas a serem tocadas ou se a música terminou. O método execute, monta a música completa (reinicia, e concatena cada linha retornada por play até a finalização da música). getNome retorna o nome da música.

A	D C# D	A	D C# D	A	D C# D
A	D C# D	God save the queen		God save the queen	
		A	D C# D	Deus salve a rainha	
Ab A	Ab A	The fascist regime		A	D C# D
A	D C# D	They made you a moron		The fascist regime	
Ab A	Ab A			o regime fascista	
		A potential h-bomb		They made you a moron	
		Ab A	Ab A	Eles fizeram de você um idiota	
		God save the queen			
		A	D C# D	A potential h-bomb	
		She ain't no human being		Uma potencial bomba h	
				Ab A	Ab A
		There is no future		God save the queen	
		Ab A	Ab A	Deus salve a rainha	
		In england's dreamland		A	D C# D
				She ain't no human being	
				Ela não é um ser humano	
				There is no future	
				não há futuro	
				Ab A	Ab A
				In england's dreamland	
				Na terra dos sonhos da Inglaterra	

Exemplo de Música - Apenas as Notas e Com a Letra Original - Original + Traduzida PT

Playlist: é a classe que representa uma playlist, composta por músicas e outras playlists. Playlist possui um método que retorna seu nome. Quando executa (método execute), a playlist retorna a concatenação de todas as músicas que a compõem, inclusive às que fazem parte das suas playlists internas, na ordem que elas foram inseridas. O método randomize, retorna uma música completa escolhida de forma pseudo-aleatória.

ATENÇÃO: PARA CADA QUESTÃO, INDIQUE OS PADRÕES DE PROJETOS UTILIZADOS E USE UM COMENTÁRIO NO TOPO DO CÓDIGO DE CADA CLASSE, INDICANDO O SEU PAPEL PARTICIPANTE NO PADRÃO IMPLEMENTADO

Sem prejuízo de outros padrões, que podem e devem ser usados para facilitar a resolução do problema serão objetos de avaliação, os padrões que resolvam os seguintes problemas:

1. Deseja-se garantir que as músicas possam ser exibidas apenas com acordes, com letras, com tradução (em diversos idiomas), com títulos, além de permitir a extensão para outros atributos, em qualquer combinação destes elementos.
2. Deseja-se garantir que o Player opere corretamente com as playlists e músicas, em qualquer combinação adicionada e para todos os PlayerModes.
3. A solução deve permitir que a mudança de modo de funcionamento do player possa ser informada por diversos meios aos possíveis diversos utilizadores

A equipe pode e deve modificar o modelo proposto para garantir a implementação das funcionalidades e dos padrões, que proporcionarão a resolução dos requisitos listados.

Normas de Entrega

1. A entrega deve ser feita mediante a disponibilização do projeto em ambiente git (github) com acesso compartilhado para o usuário pooinf008
2. A data do último commit NÃO pode ser posterior ao dia 06/07/2023 às 23:59

3. O link de compartilhamento deve ser enviado para o e-mail poo.inf008@gmail.com com o subject **[inf011-2023.1-aval2] NOME DOS PARTICIPANTES DA EQUIPE**
4. A seleção dos mails será feita por filtro, portanto atente ao subject como foi apresentado.
5. O número de componentes da equipe não pode exceder a três em nenhuma hipótese