



PCS 3111 - Laboratório de Programação Orientada a Objetos para Engenharia Elétrica

2021

Aula 09 – Programação Defensiva

Atenção

- Código inicial para resolução dos exercícios encontra-se disponível no Discord.
- Para a resolução dos exercícios, adicione atributos privados às classes conforme necessário, desde que se mantenham as assinaturas e funcionamento especificados no enunciado.
- Submeta um arquivo comprimido (faça um “.zip” – **não pode ser “.rar”**) colocando apenas os arquivos “.cpp” e “.h”. Não crie pastas no “zip”.
- Comente a função *main* ao submeter.

Exercício 01

Altere o código fornecido da classe **Playlist** seguindo as seguintes características:

1. O construtor deve jogar uma exceção do tipo `invalid_argument` (da biblioteca padrão) caso o valor de `maximoValor` seja menor ou igual a 1 ou caso a `string` `nome` seja vazia (“”). Nesse primeiro caso o argumento deve ser “`maximoValor invalido`”; e no segundo caso, “`nome invalido`”.
2. O método `adicionar` deve:
 - Não ter valor de retorno (ser `void`).
 - Gerar uma exceção do tipo `overflow_error` (da biblioteca padrão), com argumento “`musicas estourou`”, caso o vetor `musicas` esteja cheio.
 - Gerar uma exceção do tipo `invalid_argument`, com argumento “`<nome> ja adicionado`”, se a `Musica` de nome `<nome>` já tiver sido adicionada à **Playlist**;
 - Gerar uma exceção do tipo `invalid_argument`, com argumento “`Nulo`” se o ponteiro passado ao método for igual a `NULL`.

Complete a função teste com os seguintes passos na função:

1. Tente criar uma `Playlist` com `maximoValor` inválido e nome válido;
 - a. Capture a exceção, mostre a mensagem dela na tela (pule uma linha depois) e, então, delete ela;
2. Tente criar uma `Playlist` com `maximoValor` válido e nome inválido;
 - a. Capture a exceção, mostre a mensagem dela na tela (pule uma linha depois) e, então, delete ela;
3. Crie a `Playlist` de nome *Favoritos* e `maximoValor` 2;



4. Crie a Musica “*Tio Macaco*”, de duração 343;
5. Crie a Musica *Lingus*, de duração 645;
6. Crie a Musica *Kite*, de duração 372;
7. Adicione “*Tio Macaco*” à *Favoritos*;
8. Tente adicionar, mais uma vez, “*Tio Macaco*” à *Favoritos*;
 - a. Capture a exceção, mostre a mensagem dela na tela (pule uma linha depois) e, então, delete ela;
9. Tente adicionar *Lingus* e *Kite* à *Favoritos*;
 - a. Capture a exceção, mostre a mensagem dela na tela (pule uma linha depois) e, então, delete ela;

Dicas: Não se esqueça de incluir o cabeçalho `stdexcept` onde necessário. Para concatenar string, basta somá-las.

Exercício 02

Crie e implemente a exceção `SemAvaliacao` de modo que:

- Seja uma classe filha de `invalid_argument` (da biblioteca padrão).
- A classe contenha apenas seu destrutor e construtor, o qual recebe como argumento uma mensagem (string), como exposto em aula.

Modifique a classe **Musica** fornecida para que o método `getMedia` gere um erro do tipo `SemAvaliacao`, com o nome da música como argumento, caso tente-se obter a média de uma música sem avaliações – isto é, uma música na qual o método `avaliar` não foi chamado.

Refaça a função teste, para tanto considere os seguintes passos:

1. Crie a Musica “*Tio Macaco*”, de duração 343;
2. Tente calcular a média de “*Tio Macaco*”,
 - a. Capture a exceção, mostre a mensagem dela na tela (pule uma linha depois) e, então, delete ela;

Exercício 03

Na classe **Playlist**, implemente o método `getAvaliacaoDasMusicas` de forma que ele calcule a média das avaliações das músicas na **Playlist**. Via comandos try-catch, neste método, faça:



se a **Playlist** possuir uma música sem avaliações, esta música seja desconsiderada no cálculo da média das avaliações. Para isso, deve-se usar o método `getMedia` de **Musica** implementado anteriormente. Caso, ao chamar o método `getMedia` de **Musica**, aconteça uma exceção `SemAvaliacao`, capture e destrua esta exceção. Caso a **Playlist** não tenha músicas, ou nenhuma tenha avaliação, retorne 0.

Por exemplo, suponha que a **Playlist** tenha três **Musicas**: *Rockstar*, *Thunderstruck* e “*Child in Time*”. Considerando que *Rockstar* possui média 1, *Thunderstruck* possui média 5 e *Child in Time* não possui avaliações, espera-se que o método `getAvaliacaoDasMusicas` retorne o valor 3.

Refaça a função teste:

1. Crie a Playlist de nome *Favoritos* e `maximoValor` 3;
2. Crie a Musica “*Tio Macaco*”, de duração 343;
3. Avalie “*Tio Macaco*” com notas 3 e 2;
4. Crie a Musica *Lingus*, de duração 645;
5. Avalie *Lingus* com notas 4, 4 e 5;
6. Crie a Musica *Kite*, de duração 372;
7. Adicione as músicas à *Favoritos*;
8. Imprima (na função teste) a avaliação das músicas de *Favoritos*, na forma (pule uma linha depois):

`<nome> - <avaliação das músicas>`

A saída deve ser:

Favoritos – 3.41667

Testes do Judge

Exercício 1

- Playlist: exceções do construtor
- Playlist: adicionar com vetor cheio
- Playlist: adicionar com música repetida
- Playlist: adicionar NULL
- Teste da função teste

Exercício 2

- `SemAvaliacao` é filha de `invalid_argument`
- `SemAvaliacao`: `getMedia` em música sem avaliação joga `SemAvaliacao`
- Teste da função teste



Exercício 3

- Playlist: getAvaliacaoDasMusicas com todas músicas avaliadas
- Playlist: getAvaliacaoDasMusicas com música não avaliada no começo do vetor
- Playlist: getAvaliacaoDasMusicas com música não avaliada no meio do vetor
- Playlist: getAvaliacaoDasMusicas com música não avaliada no fim do vetor
- Playlist: sem músicas avaliadas
- Teste da função teste