



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

2021

Aula 03 – Conceitos Básicos de OO

Atenção

1. As definições das classes usadas nos exercícios encontram-se **disponíveis no Discord**. Use o código fornecido.
2. Os nomes, os atributos, os métodos, e as respectivas assinaturas das classes dadas **devem seguir o especificado** em cada exercício para fins de correção automática.
3. A **ordem de declaração** de atributos e métodos fornecidos **não deve ser alterada**. Caso contrário, poderá haver redução automática da nota.
4. A função main **não deve ser submetida**. Caso contrário, a correção automática retornará um *Compilation Error*.

Exercício 01

Usando a definição fornecida e as especificações a seguir, implemente a classe Musica:

```
class Musica{
public:
    string nome;
    int duracao, quantidadeDeAvaliacoes = 0, somaDasAvaliacoes = 0;

    void avaliar(int valor);
    double getMedia();

    void imprimir();
};
```

- Uma Musica possui um nome, duração (em segundos), quantidadeDeAvaliacoes e somaDasAvaliacoes;
- O método avaliar(int valor) atribui uma nota de 0 a 5 à música. A nota deve ser somada às notas dadas anteriormente e esse valor deve ser armazenado no atributo somaDasAvaliacoes (não se esqueça de incrementar a quantidadeDeAvaliacoes);
- O método getMedia() deve retornar a razão entre a soma das avaliações e a quantidade de avaliações. Como essa razão é feita por variáveis do tipo inteiro, é necessário utilizar um double para obter as casas decimais, como no exemplo a seguir:

```
int a = 5, b = 10;
double x = ((double) a) / b;
```



- O método `imprimir()` deve exibir na tela (usando o `cout`) os dados de cada `Musica` no seguinte formato:

`<nome> - <duração> segundos - <média>`

Sendo `<...>` o valor armazenado em cada variável. Por exemplo, para um objeto `Roses` com duração de 180 segundos, a saída seria (pule uma linha ao final):

`Roses - 180 segundos - 4.33333`

- Complete o código fornecido da função **teste** com os seguintes passos:
 1. Crie uma música de nome `Roses` e duração 180 segundos;
 2. Avalie `Roses` com as notas 5, 4 e 4;
 3. Imprima `Roses`.
- Não se esqueça de comentar o `main` ao submeter.

Exercício 02

Usando a definição fornecida e as especificações a seguir, implemente a classe `Playlist`:

```
class Playlist {  
public:  
    string nome;  
    Musica* m1 = NULL;  
    Musica* m2 = NULL;  
  
    int getDuracao();  
    bool adicionar(Musica* m);  
};
```

- Uma `Playlist` é constituída por um nome, e, no máximo, por duas músicas.
- O método `getDuracao()` retorna a duração total da playlist. Caso não existam músicas na `Playlist`, o método `getDuracao()` deve retornar 0.
- O método `adicionar(Musica* m)` armazena uma música em `m1` ou `m2` (nesta ordem) e retorna `true`. Caso a música já tenha sido adicionada ou não seja possível adicionar mais músicas (já existem duas músicas na `Playlist`) deve-se retornar `false`.
 - Compare as músicas usando `==`. Não deve ser possível adicionar, portanto, uma música caso o objeto já tenha sido adicionado.
 - Ao testar seu programa **não** atribua músicas diretamente pelos atributos `m1` e `m2`. Use sempre o método `adicionar()`.
- Complemente a função teste do Exercício 1 pelo acréscimo dos seguintes passos (após a impressão da música `Roses`):
 4. Crie uma música de nome `Roda-Viva` e duração 233 segundos;



5. Avalie *Roda-Viva* com as notas 5, 3, 1;
 5. Imprima *Roda-Viva* ;
 6. Acrescente as músicas *Roses* e *Roda-Viva* em uma Playlist;
 7. Imprima na saída padrão a duração da Playlist.
- Não se esqueça de comentar o main ao submeter.

Testes do Judge

Exercício 1

- Musica com nenhuma avaliação;
- Musica com uma avaliação;
- Musica com duas ou mais avaliações;
- Musica Teste getMedia;
- Musica Teste imprimir;
- Teste da função teste.

Exercício 2

- Playlist Teste getDuracao com nenhuma musica na Playlist;
- Playlist Teste getDuracao com uma musica na Playlist;
- Playlist Teste getDuracao com duas musicas na Playlist;
- Playlist Teste adicionar com nenhuma musica adicionada;
- Playlist Teste adicionar com uma musica adicionada;
- Playlist Teste adicionar com duas musicas adicionadas;
- Playlist Teste adicionar com a mesma musica;
- Teste da função teste.