



## PCS 3111 - Laboratório de Programação Orientada a Objetos para Engenharia Elétrica 2021

### Aula 11 – Biblioteca padrão

#### Atenção

- Código inicial a ser usado na resolução dos exercícios encontra-se **disponível no Discord**.
- 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

No código fornecido foram entregues as classes **Musica**, **Artista** e **Playlist**. Altere a classe **Playlist** para que ela use um **vector**, da biblioteca padrão, para armazenar as **Musicas**. Portanto, essa classe deve possuir apenas os métodos públicos apresentados abaixo.

```
Playlist(string nome);  
~Playlist();  
  
string getNome();  
int getDuracao();  
  
vector<Musica*> getMusicas();  
  
void adicionar(Musica *m);  
  
double getAvaliacaoDasMusicas() const;  
  
void imprimir();
```

Note que o método `getMusicas` mudou o tipo de retorno e o método `getQuantidadeDeMusicas` foi removido (já que ele não é mais necessário), além do parâmetro `maximoValor` no construtor. No método `adicionar` mantenha o mesmo comportamento de jogar um **invalid\_argument** caso a **Musica** já tenha sido adicionada (note que não é mais necessário verificar o espaço disponível). Adicione a **Musica** no fim do vector. Nos demais métodos mantenha o mesmo comportamento.

Não altere as demais classes (**Musica** e **Artista**).

Complete a função teste com os seguintes passos na função:

1. Crie a Playlist *Favoritos*;
2. Crie o Artista *Tom Jobim*, de quantidadeMaxima 2;



3. Adicione à *Tom Jobim* e à *Playlist Favoritos* a música *Aguas de marco*, de duração 214, avaliada com notas 5, 4 e 5;
4. Adicione à *Playlist Favoritos* a música *Mais ninguém*, de duração 168, avaliada duas vezes com a nota 3;
5. Adicione à *Tom Jobim* a música *Samba de uma nota so*, de duração 180, avaliada com notas 5 e 3;
6. Imprima a *Playlist Favoritos*;
7. Imprima *Tom Jobim*.

## Exercício 02

Altere a classe **Artista** para que ela use um **list**, da biblioteca padrão, para armazenar **Musicas**. Como não será mais necessário o argumento **quantidadeMaxima** no construtor, retire-o. Portanto, essa classe deve possuir apenas os métodos públicos apresentados a seguir.

```
Artista(string nome);  
virtual ~Artista();  
  
list<Musica*>* getMusicas();  
string getNome() const;  
  
virtual double getNota() const;  
  
bool adicionarMusica(Musica* musica);  
  
void imprimir();
```

Note que o método `getMusicas` mudou o tipo de retorno e os métodos `getQuantidadeDeMusicas` e `getQuantidadeMaxima` foram removidos. Mantenha o mesmo comportamento do método `adicionar`. Ou seja, no método `adicionar` jogue um **invalid\_argument** caso a **Musica** já tenha sido adicionada (note que não é mais necessário verificar o espaço disponível) e adicione as **Musicas** no final da estrutura. Mantenha o comportamento dos demais métodos.

Atualize a função teste do Exercício 1 para que seja compatível com as modificações feitas.

## Testes do Judge

### Exercício 1

- Playlist: adicionar Musica
- Playlist: adicionar Musica repetida no inicio do vector
- Playlist: adicionar Musica repetida no meio do vector
- Playlist: adicionar Musica repetida no fim do vector
- Teste da funcao teste



### Exercício 2

- Artista: adicionar Musica
- Artista: adicionar Musica repetida no inicio do list
- Artista: adicionar Musica repetida no meio do list
- Artista: adicionar Musica repetida no fim do list
- Teste da funcao teste