



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

2021

Aula 10 – Persistência de Objetos

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.
- 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.

Exercício 1

Implemente uma nova classe, **PersistenciaMusica**, que será responsável pela persistência dos objetos da classe **Musica** entregue (e que não deve ser alterada). Essa classe possui os seguintes métodos:

```
class PersistenciaMusica {  
public:  
    PersistenciaMusica(string arquivo);  
    virtual ~PersistenciaMusica();  
  
    void inserir(Musica* m);  
};
```

- O construtor **PersistenciaMusica** recebe o arquivo em que os dados serão salvos. Não precisa fazer nada no destrutor.
- O método **inserir** deve inserir os dados da música passada como argumento, adicionando-os ao final do arquivo.

O arquivo a ser usado por **PersistenciaMusica** deve possuir o seguinte formato:

```
nome1  
duracao1  
quantidadeDeAvaliacoes1  
somaDasAvaliacoes1  
nome2  
duracao2  
quantidadeDeAvaliacoes2  
somaDasAvaliacoes2  
...  
nomen  
duracaon  
quantidadeDeAvaliacoesn  
somaDasAvaliacoesn
```



Em um arquivo serão armazenados os dados (nome, duração, quantidadeDeAvaliacoes e somaDasAvaliacoes de uma música) de n **Musicas**. A primeira linha do arquivo deverá conter o nome da primeira **Musica**. A segunda linha deve conter a duração da primeira **Musica**. Já na terceira e na quarta linha devem existir a quantidade de avaliações e a soma das avaliações dessa mesma música. A partir da quarta linha esse padrão se repete para a segunda **Musica** e assim por diante para as n músicas. **Note que o arquivo termina com um “\n” (use o endl)!**

Exemplo: Sejam duas **Musicas**, “Valerie” e “Layla”, cujas durações sejam, respectivamente, 5 e 4 minutos. Suponha que “Valerie” tem 100 avaliações e soma 450 e que “Layla” tenha 200 avaliações e soma 990. Ao salvar “Valerie” e depois “Layla”, teremos um arquivo texto com a seguinte apresentação (em um editor de texto):

```
Valerie
5
100
450
Layla
4
200
990
```

ATENÇÃO: Assuma que o nome da **Musica** não contém espaços!

Exercício 2

Adicione à classe **PersistenciaMusica** o método

```
Musica** obter(int& quantidade);
```

que retorna um vetor (alocado dinamicamente dentro do método) com as **Musicas** armazenadas no arquivo informado no construtor da classe e a quantidade de músicas deste vetor, retornada através da variável quantidade (passada por referência). Por simplicidade, considere que o arquivo contém no máximo 10 **Musicas**. Desse modo, deve-se reconstruir todas as **Musicas** armazenadas no arquivo, com seus respectivos nomes e atributos persistidos.

Assume-se que quem chamou o método será o responsável por destruir o vetor e as **Musicas**.

Esse método deve ser implementado de modo a lidar com possíveis problemas de leitura e formatação do arquivo:

- Caso o arquivo não seja encontrado, a função deve jogar a exceção **invalid_argument**, de argumento *Erro de leitura*, da biblioteca padrão.
- Caso o arquivo esteja vazio, a função deve retornar **NULL**.
- Caso o arquivo não siga a formatação esperada ou haja uma falha de leitura, deve-se jogar a exceção **logic_error**, de argumento *Arquivo com formatacao inesperada*, da biblioteca padrão.

Complete a função teste com os seguintes passos na função:

1. Crie um objeto da classe **PersistenciaMusica**, e arquivo teste.txt;



2. Em um bloco try, faça:
 - a. Pelo método obter, obtenha um vetor de músicas do arquivo teste.txt fornecido;
 - b. Imprima: <quantidade> musica(s), em que <quantidade> é o parâmetro passado para o método obter;
 - c. Pelo método imprimir fornecido da classe Musica, imprima as músicas obtidas;
3. No catch capture a eventual exceção, imprima na saída padrão o retorno do método **what()** e, então, delete-a.

DICAS:

- Adapte o exemplo dado na apresentação (slide 27).
- Crie o vetor de **Musicas** com tamanho 10, para facilitar.
- Não se esqueça de inicializar o valor da quantidade com 0.

Testes do Judge

Exercício 1

- PersistenciaMusica testa inserir uma Musica em arquivo vazio
- PersistenciaMusica testa inserir mais de uma Musica em arquivo vazio
- PersistenciaMusica testa inserir Musicas em arquivo não vazio

Exercício 2

- PersistenciaMusica obter para arquivo vazio
- PersistenciaMusica obter para arquivo mal formatado
- PersistenciaMusica obter para arquivo inexistente
- PersistenciaMusica obter em arquivo com uma Musica
- PersistenciaMusica obter em arquivo com mais de uma Musica
- Teste da funcao teste