



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

2022

Aula 4 – Encapsulamento

Atenção

- Código inicial a ser usado na resolução dos exercícios encontra-se **disponível no e-Disciplinas**.
- Submeta um arquivo comprimido (faça um “.zip” – não pode ser “.rar”) colocando apenas os arquivos “.cpp” e “.h”. Não crie pastas no “zip”.
- Ao enviar para o Judge mantenha o `#define NUMERO_MAXIMO_VALORES 10` com o valor 10, por motivos de correção.

Utilize o código fornecido no e-Disciplinas para implementar as classes **Filme** e **Catalogo**, aplicando os conceitos de encapsulamento vistos em aula.

Exercício 1

No arquivo teste.cpp é fornecida a definição e a implementação da classe Filme. Separe-a em dois arquivos, “**Filme.h**” e “**Filme.cpp**”. O arquivo .h deve conter apenas a definição. O arquivo .cpp deve conter apenas a implementação. Use adequadamente as diretivas de compilação.

Implemente os métodos `getNome`, `setNome`, `getDuracao` e `setDuracao` apropriadamente (retorne o valor do respectivo atributo no `get` e defina o valor do respectivo atributo usando o parâmetro do `set`). Defina a visibilidade dos atributos e dos métodos de modo que os atributos sejam acessíveis **apenas** pelos métodos da classe e os métodos sejam acessíveis externamente.

No arquivo teste.cpp deixe apenas as funções `teste1`, `teste2` e o `main`.

Complete o código fornecido, para tanto considere os seguintes passos na função `teste1`:

1. Crie um Filme de nome *Bao* e duração 8 (representando 8 minutos);
2. Imprima *Bao*.

Exercício 2

Implemente a classe **Catalogo**, fornecida. Note que essa classe utiliza a classe **Filme**, do Exercício 1. Novamente, separe a classe em dois arquivos, “**Catalogo.h**” e “**Catalogo.cpp**”, usando adequadamente as diretivas de compilação. Defina corretamente a **visibilidade** de seus métodos e atributos.

A implementação da classe **Catalogo** deve atender aos seguintes requisitos:

- O método `getDuracaoTotal` deve retornar o somatório da duração de cada Filme no **Catalogo**. Caso não existam filmes adicionados ao catálogo, esse método deve retornar 0.



- **Dica:** Use o método da classe Filme.
- O método adicionar deve adicionar, se possível, o objeto do tipo Filme passado como parâmetro ao vetor **filmes** (crie este atributo). Caso o vetor já esteja completamente preenchido **ou o objeto já tenha sido adicionado** (compare o objeto com ==), o método não modifica o vetor e retorna **false**. Caso seja bem-sucedido, deve retornar **true**. Utilize a constante NUMERO_MAXIMO_VALORES como o número máximo de filmes que o catálogo comporta. Controle a quantidade de objetos adicionados com um atributo quantidade, ou seja, caso exista apenas um filme no catálogo, a quantidade será igual a 1; caso existam dois filmes, a quantidade será igual a 2 (**use esse atributo para saber em qual posição do vetor deve ser adicionado o valor**).
- O método getQuantidade retorna o número de filmes que foram adicionadas ao vetor filmes, através do atributo quantidade.
- O método imprimir deve mostrar na tela as informações do catálogo, seguindo o formato (pulando uma linha no final):

Catalogo com <quantidade> filmes - <duração total> minutos no total

Além disso, imprima as informações de todos os filmes que foram adicionados ao catálogo (usando o método imprimir de Filme). Por exemplo:

Catalogo com 2 filmes - 430 minutos no total
E o Vento Levou - 240 minutos
Titanic - 190 minutos

- Implemente a função teste2, executando apenas os seguintes passos:
 1. Crie um Filme de nome *O Magico de Oz* e duração 100 minutos;
 2. Crie outro Filme de nome *Alien* e duração 120 minutos;
 3. Crie um Catalogo;
 4. Adicione *O Magico de Oz* e *Alien* ao Catalogo;
 5. Imprima o Catalogo.

Dica: a classe Filme já possui um método imprimir – utilize-o para facilitar a implementação do método imprimir de Catalogo!

Lembre-se de utilizar a diretiva `#ifndef` nos arquivos de cabeçalhos (".h") para evitar problemas de conflitos de definição de classes causados por múltiplas inclusões de um mesmo cabeçalho.

Dicas importantes

1. Os nomes, os atributos, os métodos, e as respectivas assinaturas das classes dadas **devem seguir exatamente o especificado** para fins de correção automática.
2. A função main não deve ser submetida. Caso contrário, a correção automática retornará um *Compilation Error*.



Testes do Judge

Exercício 1

- Teste dos *setters* e *getters* de Filme;
- Teste do método imprimir de Filme;
- Teste da função teste1.

Exercício 2

- Teste adicionar com vetor vazio;
- Teste adicionar com vetor parcialmente preenchido;
- Teste adicionar com vetor cheio;
- Teste adicionar filme repetido;
- Teste getQuantidade de Catalogo;
- Teste getDuracaoTotal com vetor vazio;
- Teste getDuracaoTotal com vetor parcialmente preenchido;
- Teste getDuracaoTotal com vetor cheio;
- Teste imprimir sem filmes;
- Teste imprimir com 1 filme;
- Teste imprimir com vários filmes;
- Teste da função teste2.