



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

2022

Aula 03 – Conceitos Básicos de OO

Atenção

1. As definições das classes usadas nos exercícios encontram-se **disponíveis no e-Disciplinas**. 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 a especificação a seguir, implemente a classe Filme:

```
class Filme {  
public:  
    void assistir(int tempo);  
  
    void imprimir();  
  
    string nome;  
    int duracao;  
    int visualizacoes = 0;  
};
```

- Um Filme possui, como atributos, um nome, uma duração em minutos e a sua quantidade de visualizações;
- O método assistir recebe como argumento o valor do tempo em que o filme foi assistido por uma pessoa, devendo incrementar em uma unidade o número de visualizações do filme **caso** o valor de tempo seja maior ou igual a 70% da duração do filme. Como será necessário obter a porcentagem do filme que foi assistida, 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;
```



Por exemplo, considere o Filme *Mad Max* com 0 visualizações e duração de 88 min, para o qual chamaremos o método `assistir`. Após chamar `assistir` com 50, o atributo `visualizacoes` deve continuar com 0. Após chamar `assistir` com 80, o atributo deve ficar com 1. Ao chamar `assistir` com 88, o atributo é incrementado em 1 unidade, passando a valer 2.

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

Filme: <nome> - <duração> minutos - <visualizações> visualizacoes

Sendo <...> o valor armazenado em cada variável. Por exemplo, para um filme com nome *Vingadores* com duração de 150 minutos e com 5 visualizações, a saída seria (pule uma linha ao final):

Filme: Vingadores - 150 minutos - 5 visualizacoes

- Implemente a função **teste1** declarada no código fornecido seguindo os próximos passos:
 1. Crie um filme de nome *Mad Max* e duração 120 minutos;
 2. Chame o método `assistir` informando 100 min;
 3. Chame o método `assistir` informando 87 min;
 4. Imprima o filme.
- Não se esqueça de comentar a `main` ao submeter.

Exercício 02

Usando a definição fornecida e a especificação a seguir, implemente a classe `Catalogo`:

```
class Catalogo {
public:
    int getDuracaoTotal();
    int getTotalDeVisualizacoes();
    bool adicionar(Filme* f);

    Filme* f1 = NULL;
    Filme* f2 = NULL;
    int quantidade = 0;
};
```



- Um Catalogo é constituído por, no máximo, dois filmes. O atributo `quantidade` controla o número de filmes que foram adicionados ao catálogo;
- O método `getDuracaoTotal` retorna a duração total do catálogo. Caso não existam filmes no Catalogo, este deve retornar 0;
- O método `getTotalDeVisualizacoes()` retorna a soma das visualizações dos filmes componentes do catálogo. Caso não existam filmes, retorna-se 0;
- O método `adicionar` armazena um filme em `f1` ou `f2` (nesta ordem) e retorna `true`. Use o atributo `quantidade` para saber em qual das variáveis deve-se armazenar o filme (se 0, é em `f1`; se 1, é em `f2`). Caso o filme já tenha sido adicionado ou não seja possível adicionar mais filmes (já existem dois filmes no Catalogo), deve-se retornar `false`.
 - Ao testar seu programa, **não** atribua filmes diretamente pelos atributos `f1` e `f2`. Use sempre o método `adicionar`.
- Implemente a função **teste2** declarada no código fornecido seguindo os próximos passos:
 1. Copie o que foi feito em **teste1** para o **teste2**;
 2. Crie um filme de nome *Mercenarios* e duração 103 minutos;
 3. Chame o método `assistir` de *Mercenarios* passando os valores 103, 90 e 98;
 4. Imprima *Mercenarios*;
 5. Crie um catálogo e adicione (usando o método `adicionar`) os filmes *Mad Max* e *Mercenarios*, nesta ordem;
 6. Imprima a duração total e o total de visualizações do catálogo criado, em linhas separadas;
- Não se esqueça de comentar o `main` ao submeter no *judge*.

Testes do Judge

Exercício 1

- Filme nunca assistido;
- Filme Teste assistir com menos de 70% de sua duração assistida;
- Filme Teste assistir com exatamente 70% de sua duração assistida;
- Filme Teste assistir com mais de 70% de sua duração assistida;
- Filme Teste assistir mais de uma vez com 0 visualizações;
- Filme Teste assistir mais de uma vez com 1 visualização;
- Filme Teste assistir mais de uma vez com 2 visualizações;
- Filme Teste imprimir;
- Teste função `teste`.



Exercício 2

- Catalogo sem nenhum filme;
- Catalogo Teste `getDuracaoTotal` com um filme;
- Catalogo Teste `getDuracaoTotal` com dois filmes;
- Catalogo Teste `getTotalDeVisualizacoes` com um filme;
- Catalogo Teste `getTotalDeVisualizacoes` com dois filmes;
- Catalogo Teste `adicionar` com nenhum filme adicionado;
- Catalogo Teste `adicionar` com um filme distinto adicionado;
- Catalogo Teste `adicionar` com dois filmes adicionados;
- Catalogo Teste `adicionar` com um filme já adicionado;
- Teste da função teste.