



## PCS 3111 - Laboratório de Programação Orientada a Objetos para Engenharia Elétrica 2022

### Aula 09 – Programação Defensiva

#### Atenção

- Código inicial para resolução dos exercícios encontra-se disponível no e-Disciplinas.
- Para a resolução dos exercícios, adicione atributos privados às classes conforme necessário, desde que se mantenham as assinaturas e funcionamento especificados no enunciado.
- 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.

**AVISO:** para evitar problemas de compilação no Judge, envie os exercícios à medida que implementá-los!

#### Exercício 01

Altere o código fornecido da classe **Filme** seguindo as seguintes características:

1. O construtor deve jogar uma exceção do tipo `invalid_argument` (da biblioteca padrão) caso o valor de `duracao` seja menor ou igual a zero. O argumento da exceção deve ser “`duracao invalida`”.
2. O método `assistir` deve jogar uma exceção do tipo `invalid_argument` caso o tempo seja negativo ou maior do que a duração do filme. O argumento da exceção deve ser “`tempo invalido`”.

Complete a função `teste1` com os seguintes passos na função:

1. Tente criar um `Filme` com `duracao -1`;
  - a. Capture a exceção, mostre a sua mensagem na tela (pule uma linha depois) e, então, delete-a;
2. Crie o `Filme` de nome *Matrix* e `duracao 136` minutos;
3. Tente chamar o método `assistir` com tempo negativo;
  - a. Capture a exceção, mostre a sua mensagem na tela (pule uma linha depois) e, então, delete-a;
4. Tente chamar o método `assistir` com tempo de 137 minutos;
  - a. Capture a exceção, mostre a sua mensagem na tela (pule uma linha depois) e, então, delete-a;



**Dicas:** Não se esqueça de incluir o cabeçalho `stdexcept` onde necessário.

### Exercício 02

Crie e implemente a exceção `SemVisualizacoes` de modo que:

- Seja uma classe filha de `logic_error` (da biblioteca padrão).
- A classe contenha apenas seu destrutor e construtor, o qual recebe como argumento uma mensagem (`string`), como exposto em aula.

Modifique a classe **Catalogo** fornecida para que o método `getFilmeMaisVisto` gere uma exceção do tipo `SemVisualizacoes` caso todos os filmes do catálogo possuam 0 visualizações (isso inclui quando o catálogo não possuir filmes). O argumento das exceções deve ser “`catalogo sem visualizacoes`”.

Complete a função `teste2`, para tanto considere os seguintes passos:

1. Crie o Filme de nome *Matrix* e duracao 136 minutos;
2. Crie o Filme de nome *Smurfs* e duracao 100 minutos;
3. Crie um Catalogo com quantidadeMaxima 2 e adicione os filmes criados na ordem de criação;
4. Tente chamar o método `getFilmeMaisVisto` do catálogo criado;
  - a. Capture a exceção, mostre a sua mensagem na tela (pule uma linha depois) e, então, delete-a;

### Exercício 03

A classe **Plano** pode conter vários catálogos diferentes. Nessa classe, implemente o método `getDestaque` de forma que ele retorne o **Filme** mais visto dentre todos os filmes de todos seus catálogos. Via comandos `try-catch`, neste método, faça: se o **Catalogo** não possuir visualizações, desconsidere-o na busca pelo filme em destaque. Para isso, deve-se usar o método `getFilmeMaisVisto` de **Catalogo** implementado anteriormente. Caso, ao chamar o método `getFilmeMaisVisto` de **Catalogo**, aconteça uma exceção `SemVisualizacoes`, capture e destrua esta exceção. Caso o **Plano** não possua catálogos, ou todos seus catálogos tenham zero visualizações, retorne nulo.

Complete a função `teste3`:



1. Crie um Plano de quantidadeMaxima 2;
2. Crie dois Catalogos de quantidadeMaxima 2;
3. Adicione os catálogos ao Plano;
4. Crie o Filme *Matrix* com duração 136;
5. Crie o Filme *Smurfs* com duração 100;
6. Crie o Filme *O Regresso* com duração 156;
7. Crie o Filme *Duna* com duração 160;
8. Adicione a um catálogo os filmes *Matrix* e *Smurfs* e ao outro os filmes *O Regresso* e *Duna*, respectivamente;
9. Chame o método `getDestaque` do Plano (e ignore o retorno);
10. Faça com que *Matrix* tenha três visualizações, *Smurfs* uma visualização, *O Regresso* duas visualizações e *Duna* zero visualizações;
11. Imprima (usando o método `imprimir`) o filme em destaque do plano;

## Testes do Judge

### Exercício 1

- Filme: exceções do construtor
- Filme: assistir com tempo negativo
- Filme: assistir com tempo maior do que o filme
- Teste da função `teste1`

### Exercício 2

- `SemVisualizacoes` é filha de `logic_error`
- `SemVisualizacoes: Catalogo` sem visualizações joga `SemVisualizacoes`
- Teste da função `teste2`

### Exercício 3

- Plano: `getDestaque` com todos os catálogos com visualizações;
- Plano: `getDestaque` com catálogo sem visualizações no começo do vetor do Plano
- Plano: `getDestaque` com catálogo sem visualizações no meio do vetor do Plano
- Plano: `getDestaque` com catálogo sem visualizações no fim do vetor do Plano
- Plano: todos os catálogos sem visualizações
- Teste da função `teste3`