



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

2022

Aula 11 – Namespaces, Templates e Biblioteca Padrão

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*.

Considere as classes **Filme** e **Catalogo**, implementadas nas aulas anteriores e que são fornecidas para esta aula.

Exercício 01

Até agora, vínhamos trabalhando com vetores para armazenar coleções em uma classe. Agora, queremos trabalhar com novas estruturas de dados, como os *containers* vistos nesta aula. Para isso, modifique a classe **Catalogo** fornecida para que os seus filmes sejam armazenados em um container do tipo `vector`, e não mais em um vetor simples. Os métodos públicos da classe devem ficar como a seguir:

```
Catalogo(string nome);  
virtual ~Catalogo();  
  
string getNome();  
void adicionar(Filme *f);  
int getDuracaoTotal();  
vector <Filme*> getFilmes();  
  
void imprimir();
```

- No construtor, aloque dinamicamente o `vector` que será utilizado para armazenar os ponteiros;
- No destrutor, destrua o `vector` de filmes alocado dinamicamente (mas não destrua os filmes);



- O método `getNome` deve retornar o nome do catálogo (não é necessário alterá-lo);
- No método `adicionar`, se o filme que se deseja adicionar já está presente no catálogo, jogue uma exceção do tipo `invalid_argument`. Caso contrário, adicione o filme. Note que não existe mais o problema de *overflow* para a estrutura do vector.
- Os métodos `getDuracaoTotal`, `getFilmes` e `imprimir` devem possuir os mesmos comportamentos fornecidos no código, mas devem ser adaptados para o uso de vector;
- Implemente a função `teste1()` conforme os passos a seguir:
 1. Crie um catálogo de nome *Sci-Fi*;
 2. Crie o filme *Star Wars* de 120 minutos de duração e adicione-o ao catálogo;
 3. Crie o filme *Planeta dos Macacos* de 90 minutos de duração e adicione-o ao catálogo;
 4. Imprima o catálogo;
 5. Delete o catálogo e os filmes criados.

Exercício 02

Implemente a classe **Plano**, que representa um plano de assinatura que cobre alguns catálogos de filmes. Ela recebe, em seu construtor, uma string que define o tipo de plano (por exemplo, “Basico”, “Premium”, etc). Seus métodos públicos encontram-se a seguir:

```
Plano(string tipo);  
virtual ~Plano();  
  
void adicionar(Catalogo *c);  
list <Catalogo*> getCatalogos();  
  
void imprimir();
```

- No construtor, aloque dinamicamente a `list` de catálogos do plano e delete-a no destrutor (mas não destrua os catálogos adicionados);
- No método `adicionar`, caso o nome do catálogo passado como argumento já exista no plano, jogue uma exceção do tipo `invalid_argument` (a mensagem de erro passada para a exceção não será analisada). Caso contrário, adicione o catálogo ao final da lista.



Note que você deve utilizar o nome do catálogo para a comparação, e não o ponteiro do catálogo em si, pois é possível que haja catálogos diferentes com nomes iguais.

- O método `getCatalogos` deve retornar o ponteiro para a `list` de catálogos do plano;
- No método `imprimir`, a primeira linha impressa deve ser:

Plano <tipo> - <número de catálogos> catalogos

Pule uma linha após imprimir isso. Em seguida, para cada catálogo no plano, chame o seu método `imprimir`, o qual foi adaptado no exercício 01.

- Implemente a função `teste2()` conforme os passos a seguir:
 1. Crie um plano de tipo `Basico`;
 2. Repita os passos de 1 a 3 do `teste1`;
 3. Crie um novo catálogo de nome *Guerra*;
 4. Crie o filme *A Lista de Schindler* de 120 minutos de duração e adicione-o ao catálogo *Guerra*;
 5. Adicione os catálogos *Sci-Fi* e *Guerra*, nessa ordem, ao plano;
 6. Imprima o plano;
 7. Delete o plano, os catálogos e os filmes criados.

Testes do Judge

Exercício 1

- Catalogo Teste `getFilmes`;
- Catalogo Teste adicionar filmes distintos;
- Catalogo Teste adicionar um filme já existente no catálogo;
- Catalogo Teste `getDuracaoTotal` adaptado;
- Catalogo Teste `imprimir`.
- Teste da função `teste1`.

Exercício 2

- Plano Teste adicionar catálogos com nomes distintos;
- Plano Teste adicionar catálogo com nome já existente;
- Plano Teste `imprimir`;
- Teste da função `teste2`.