



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

2022

Aula 10 – Persistência de Objetos

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”.
- 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, **PersistenciaDeFilme**, que será responsável pela persistência dos objetos da classe **Filme** entregue (e que não deve ser alterada). Essa classe possui os seguintes métodos:

```
class PersistenciaDeFilme {  
public:  
    PersistenciaDeFilme(string arquivo);  
    virtual ~PersistenciaDeFilme();  
  
    void inserir(Filme* filme);  
};
```

- O construtor de **PersistenciaDeFilme** recebe uma string contendo o nome do arquivo em que os dados serão salvos. O destrutor não precisa fazer nada.
- O método **inserir** deve inserir no arquivo informado no construtor os dados do filme passado como argumento, adicionando-o ao final do arquivo.

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

```
nome1  
duracao1  
visualizacoes1  
nome2  
duracao2  
visualizacoes2  
...  
nomen  
duracaon  
visualizacoesn
```



Em um arquivo serão armazenados os dados (nome, duração e visualizações) de n **Filmes**. A primeira linha do arquivo deverá conter o nome do primeiro **Filme**, a segunda linha deve conter a sua duração e, por fim, a terceira linha deve conter o seu número de visualizações. A partir da quarta linha esse padrão se repete para o segundo **Filme** e assim por diante para os n próximos filmes. **Note que o arquivo termina com um “\n” (use o endl)!**

Exemplo: Sejam dois **Filmes**, “Moana” e “Interestelar”, cujas durações sejam, respectivamente, 103 e 169 minutos. Suponha que “Moana” tenha 100 visualizações e que “Interestelar” tenha 200. Ao salvar “Moana” e depois “Interestelar”, teremos um arquivo texto com a seguinte apresentação (em um editor de texto):

```
Moana
103
100
Interestelar
169
200
```

Execute os seguintes passos na função teste1:

1. Crie dois filmes: “Luca” de 95 minutos com 12 visualizações e “Tenet” de 150 minutos e com 7 visualizações;
2. Crie também um objeto **PersistenciaDeFilme** que deve persistir os dados no arquivo *teste1.txt*;
3. Insira os dois filmes criados, primeiro “Luca” e depois “Tenet”.
4. Delete todos os objetos alocados dinamicamente.

OBS: Por facilidade, a classe **Filme** fornecida possui um segundo construtor, em que é possível atribuir um valor inicial para o atributo *visualizacoes*, ou seja, não é necessário chamar o método *assistir* determinado número de vezes.

Exercício 2

Adicione à classe **PersistenciaDeFilme** o método

```
Catalogo* obter();
```

que retorna um novo **Catalogo** com os **Filmes** armazenados no arquivo informado no construtor da classe. Por simplicidade, considere que o arquivo contém no máximo 10 **Filmes**. Desse modo, deve-se reconstruir todos os **Filmes** armazenados no arquivo, seguindo o formato explicado no exercício anterior.

Assume-se que quem chamou o método será o responsável por destruir o **Catalogo**.

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

- Caso o arquivo não seja encontrado, o método 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.

ATENÇÃO: Assuma que o nome do **Filme** não contém espaços!

Complete a função teste2 com os seguintes passos na função:

1. Crie um objeto da classe **PersistenciaDeFilme**, passando o nome *teste2.txt* (o arquivo teste2.txt é fornecido, mas o Judge usa um outro) como argumento;
2. Em um bloco try, faça:
 - a. Pelo método obter, obtenha o catálogo referente aos filmes do arquivo *teste2.txt*;
 - b. Chame o método imprimir do catálogo, se for diferente de NULL
 - c. Delete o catálogo
3. No catch capture a eventual exceção, imprima na saída padrão o retorno do método what() (pulando uma linha ao final) e, então, delete-a.
4. Delete o objeto de persistência.

Testes do Judge

Exercício 1

- PersistenciaFilme inserir um Filme em arquivo vazio
- PersistenciaFilme inserir mais de um Filme em arquivo vazio
- PersistenciaFilme inserir Filmes em arquivo não vazio
- Teste da funcao teste1

Exercício 2

- PersistenciaFilme obter para arquivo vazio
- PersistenciaFilme obter para arquivo mal formatado
- PersistenciaFilme obter para arquivo inexistente
- PersistenciaFilme obter em arquivo com um Filme
- PersistenciaFilme obter em arquivo com mais de um Filme
- Teste da funcao teste2