

## Exercício de Programação Orientada a Objetos

1- Você foi encarregado de construir um pequeno sistema para uma Locadora de Filmes, esse sistema deverá conter três classes: Filme, Cliente e Locadora.

### Classe Filme:

Atributos:

**nome** : (String)

**classificacao** : (int) → poderá assumir o valor de 18,16,12 ou -1 . O -1 indica que a classificação é livre.

**codigo**: (int)

**emprestado**: (boolean) → esse atributo serve para indicar se o livro está ou não emprestado no momento.

**categoria**: (Char) poderá assumir o valor de A, B ou C.

**Métodos**: Deverão ser construídos métodos para manipulação dos atributos (gets e sets). Caso não seja fornecido o valor boolean emprestado, deverá ser considerado como false.

### Classe Cliente:

Atributos:

**nome** : (String)

**idade**: (int)

**codigo**: (int)

**endereço** : (String)

**telefone**: (String)

**Métodos**: Deverão ser construídos métodos para manipulação dos atributos (gets e sets). Um cliente só poderá ser cadastrado se pelo menos os atributos nome, idade ,telefone e codigo forem fornecidos no momento

do cadastro, caso o endereço não seja fornecido, deverá ser inicializado como uma string vazia "".

## **Classe Locadora:**

**nome:** (String)

**cliente1,cliente2,cliente3:** (Cliente)

**filme1,filme2,filme3:** (Filme)

### **Métodos:**

**Construtor:** → O construtor recebe como parâmetro o nome da locadora.

**getNome:** → retorna o nome da locadora.

**setNome:** → recebe como parâmetro uma String e atualiza o nome da locadora.

**Cadastro de Clientes:** → recebe como parâmetro os dados do cliente, faz a validação do código (verifica se existe algum cliente cadastrado com aquele código, caso exista imprime uma mensagem na tela de erro “Não foi possível cadastrar o cliente”).

Deve-se cadastrar na ordem: cliente1, cliente2 e cliente3, caso os três clientes já estejam cadastrados deve-se imprimir uma mensagem de erro: “Os três clientes já foram cadastrados”.

Obs: Deve-se verificar se as referências para clientes não são nulas, antes de verificar o código do cliente.

**Cadastro de Filmes:** → recebe como parâmetro os dados do filme, faz a validação do código (verifica se existe algum filme cadastrado com aquele código, caso exista imprime uma mensagem na tela de erro “Não foi possível cadastrar o filme”).

Deve-se cadastrar na ordem: filme1, filme2 e filme3, caso os três filmes já estejam cadastrados deve-se imprimir uma mensagem de erro: “Os três filmes já foram cadastrados”.

Obs: Deve-se verificar se as referências para filmes não são nulas, antes de verificar o código do filme.

**Listagem de todos os filmes da locadora:** -> esse método lista o nome dos filmes cadastrados na locadora.

Obs: Deve-se verificar se as referências para filmes não são nulas.

**Listagem de filmes por categoria:** → recebe como parâmetro um char e imprime na tela o nome dos filmes associados a essa categoria.

Obs: Deve-se verificar se as referências para filmes não são nulas.

**Listagem de filmes por classificação:** → recebe como parâmetro um int e imprime na tela o nome dos filmes associados a essa classificação.

Obs: Deve-se verificar se as referências para filmes não são nulas.

**Listagem de filmes que não estejam emprestados:** esse método lista o nome dos filmes que não estão emprestados.

Obs: Deve-se verificar se as referências para filmes não são nulas.

**Consulta de Cliente:** → recebe como parâmetro um código e retorna o cliente, caso esteja cadastrado e null caso contrário.

Obs: Deve-se verificar se as referências para clientes não são nulas.

**Consulta de Filme:** → recebe como parâmetro um código e retorna o filme, caso esteja cadastrado e null caso contrário.

Obs: Deve-se verificar se as referências para filmes não são nulas.

**Empréstimo de Filme:** → recebe como parâmetro o código do cliente e o código do filme, após isso o método deve fazer a verificação se existem cliente e filme com os respectivos códigos, caso exista ele verifica se o filme está ou não emprestado e se o cliente tem idade para o empréstimo. Se todas as condições para empréstimo forem satisfeitas, o método deverá imprimir na tela:

“Empréstimo realizado com sucesso

Nome do Filme: (Nome do Filme)  
Nome do Cliente: (Nome do Cliente)  
Valor: (valor) ”

Obs.: O valor do empréstimo para as categorias A,B e C é respectivamente de R\$ 5,00, R\$ 3,50 e R\$ 2,00

**Devolução de Filme:** → recebe como parâmetro o código do filme, caso o filme esteja cadastrado e emprestado, ele atualiza a condição de emprestado para false e imprime a mensagem: “Filme: (Nome do Filme) devolvido com sucesso”. Caso não exista filme com este código, deve-se imprimir: “Filme não existente na base de Dados”. Caso exista filme com este código, mas o mesmo não está emprestado, deve-se imprimir: “Filme: (Nome do Filme) não está emprestado.”

**O sistema deverá ser implementado no BlueJ:**

**Data de Entrega: 24/09/2010**