

Este exercício pode ser discutido com os colegas ou o professor, mas a entrega deve ser individual.

Criar uma pequena aplicação que responda, e teste, o que é solicitado.

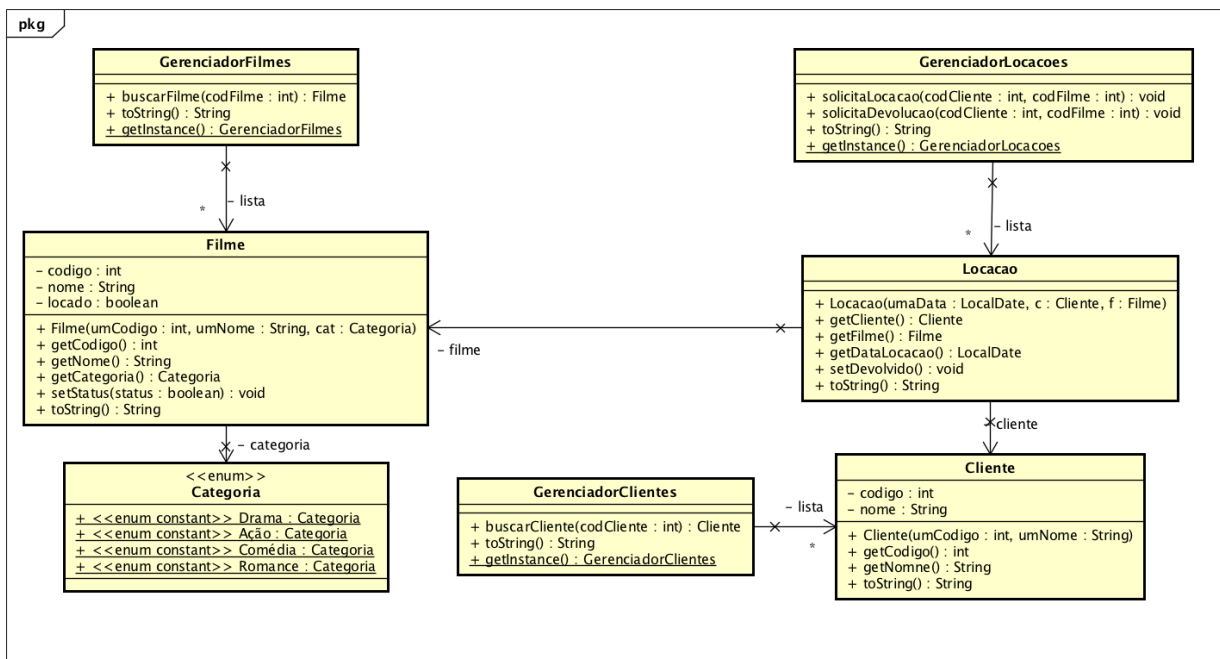
Este exercício é OPCIONAL

- substitui a nota de um exercício não entregue ou,
- se todos foram entregues, complementa a nota de trabalho(s) entregue(s) incompleto(s).

Entrega

- em formato “zip” na sala Moodle disponibilizada.

Uma antiga loja de locação resiste aos novos tempos e mantém um serviço de locação de filmes. Os dados são armazenados em um cadastro de filmes, clientes e locações, conforme o diagrama abaixo. Os filmes estão agrupados nas seguintes categorias: Drama, Ação, Comédia e Romance. Note que a hierarquia de classes apresentada está bastante simplificada: há atributos, métodos e parâmetros faltando e você deverá incluí-los caso sejam necessários para responder às questões a seguir.



1. **(2,0 pt)** Apresente a implementação do método *solicitaLocacao* da classe *GerenciadorLocacoes*. Muitas vezes os códigos recebidos são inválidos e então o método deve validar os valores recebidos. Se o código do cliente ou do filme não existirem nos cadastros, o método deverá gerar uma exceção do tipo *IllegalArgumentException* informado qual o erro ocorrido, caso contrário uma nova instância de locação é gerada e adicionada ao cadastro.

Dica: note no diagrama o método *solicitaLocacao* recebe dois inteiros (para os códigos de cliente e filme) e as locações possuem referencias para os objetos cliente e filme.

2. Explorando obrigatoriamente operações de *streams* com expressões lambda, escreva os seguintes métodos (você deve analisar e decidir em qual das classes cada um dos métodos deve ser colocado):
 - a) **(1 pt)** Mostrar o nome de todos os filmes assistidos por um determinado cliente;
 - b) **(1 pt)** Retornar o número total de filmes locados de uma determinada categoria.
 - c) **(1 pt)** Escrever o nome de todos os clientes que assistiram um determinado filme. Os nomes devem ser apresentados em ordem crescente.
3. **(1 pt)** Escreva um método que retorne um dicionário contendo, para cada categoria o número de clientes que locaram filmes daquela categoria. Você pode decidir se categorias que não tiveram filmes locados aparecem, ou não, no dicionário (como você preferir fazer).
4. **(1 pt)** Explorando obrigatoriamente operações de conjuntos, escreva um método que retorne um Set com o nome de todos os filmes que nunca foram locados.
5. **(1 pt)** Escreva um método que salve o arquivo de filmes em um arquivo, em formato csv.
6. **(2,0 pt)** Escreva uma pequena aplicação (sem interface, apenas o método *main*) que permita testar cada um dos exercícios solicitados.